A Matrix-Based Heuristic Algorithm for Extracting Multiword Expressions from a Corpus

Orhan Bilgin
Lancaster University, Linguistics and English Language Department
County South, Lancaster University, Lancaster, UK, LA1 4YL
orhan@zargan.com

Abstract

This paper describes an algorithm for automatically extracting multiword expressions (MWEs) from a corpus. The algorithm is node-based, i.e. extracts MWEs that contain the item specified by the user, using a fixed window-size around the node. The main idea is to detect the frequency anomalies that occur at the starting and ending points of an ngram that constitutes a MWE. This is achieved by locally comparing matrices of observed frequencies to matrices of expected frequencies, and determining, for each individual input, one or more sub-sequences that have the highest probability of being a MWE. Top-performing sub-sequences are then combined in a score-aggregation and ranking stage, thus producing a single list of score-ranked MWE candidates, without having to indiscriminately generate all possible sub-sequences of the input strings. The knowledge-poor and computationally efficient algorithm attempts to solve certain recurring problems in MWE extraction, such as the inability to deal with MWEs of arbitrary length, the repetitive counting of nested ngrams, and excessive sensitivity to frequency. Evaluation results show that the best-performing version generates top-50 precision values between 0.71 and 0.88 on Turkish and English data, and performs better than the baseline method even at n=1000.

Keywords: multiword expression, MWE, phraseology, extraction, ngram, observed frequency, expected frequency, Turkish, English

1. Introduction

Multiword expressions (MWEs) are conventionalized word combinations such as *at the expense of …*, *good morning*, *execute an agreement*, *31 January 2016*, *United Nations Children's Fund*, or *the proverbial elephant*. They are complex structures that contain syntactic, morphological, phonological, semantic, pragmatic, and discourse-functional information (Croft and Cruse, 2004, p. 258) and behave as single units of meaning (Sinclair, 2004, p. 39).

MWEs have been defined in terms of their non-compositionality (Villavicencio et al., 2005), lexical, syntactic and semantic idiosyncrasy (Sag et al., 2002; Baldwin and Kim, 2010; Mel'čuk, 1998), lexicalization (Wray, 2009; Maziarz, Szpakowicz, and Piasecki, 2015) semantic unity (Moon, 1998; Calzolari et al., 2002), syntactic unity (Kjellmer, 1987; Dias, 2003), institutionalization (Pawley and Syder, 1983), pragmatic specialization (Siepmann, 2005) and frequency (Grant and Bauer, 2004; Gries, 2008), among others. This diversity of approaches probably reflects the inherently complex nature of the phenomenon (Wray and Perkins, 2000, p. 3; Schmitt and Carter, 2004, p. 2).

MWEs are numerous; Jackendoff (1997) estimates that they number on about the same order of magnitude as individual words (p. 156). They are frequent; Erman and Warren (2000) report that on average they make up 55% of spoken and written language (p. 37). In view of this pervasiveness, a *MWE lexicon*, i.e. a classified inventory of habitually co-occurring lexical items, is an essential component of the description of any language (Mel'čuk, 2006, p. 3; Moon, 2008, p. 314). It is also important for natural language processing (NLP) and related disciplines, where MWEs still are an unsolved problem (Shwartz and Dagan, 2019; Nivre, 2021, p. 99). Despite the recent success of deep learning models in various NLP tasks, at least some of the performance issues faced by end-to-end pipelines like *Stanza* (Qi et al., 2020) and *UDPipe* (Straka and Straková, 2017) and the systems that use them seem to be caused by the following facts: (a) they use individual words as a unit of analysis, despite convincing evidence that "the normal primary carrier of meaning is the phrase and not the word" (Sinclair, 2008, p. 409), and (b) they rely on a strict separation of the lexical, morphological, syntactic, and semantic levels, ignoring the ubiquity of MWEs, which can be viewed as "data structure[s] that [integrate] all possible kinds of linguistic information in a single representation" (Trijp, 2018). The solution might lie in developing more complex data structures that recognize the existence of a phraseological level that crosses word boundaries and cuts across the traditional levels of analysis. MWE lexicons are essential linguistic resources in this regard.

Because unaided speakers cannot reliably discover significant recurring patterns in their native language through conscious reflection (Church et al., 1991, p. 1; Stubbs, 2002, p. 219), MWE lexicons must be created automatically or semi-automatically, using large amounts of usage data. The task of *MWE extraction*, then, can be defined as "a process that takes as input a text and generates a list of MWE candidates, which can be further filtered by human experts before their integration into lexical resources." (Constant et al., 2017, p. 847)

A large number of methods have been proposed for the automatic extraction of MWEs from corpora during the last fifty years (Section 2). Most of the focus has been on resource-rich Indo-European languages like English, German and French. This paper reports on an effort to develop a MWE extraction algorithm that requires as little linguistic knowledge as possible. Although the algorithm was primarily designed for Turkish, a language whose complex morphology has proven to be challenging for NLP (Oflazer, 2014, p . 639), preliminary results show that it performs equally well on English data (Section 4.3), suggesting that it is to some extent language-independent.

After discussing existing methods in Section 2, I will describe the proposed algorithm in Section 3, present the results of an experiment to evaluate its performance in Section 4, and discuss results and make concluding remarks in Section 5.

Proceedings of the 18th Workshop on Multiword Expressions (MWE 2022) n@LREC2022, pages 37–48

Marseille, 25 June 2022. © European Language Resources Association (ELRA), licensed under CC-BY-NC-4.0

2. Existing Extraction Algorithms

The majority of MWE extraction algorithms are based on the statistical manipulation of *ngrams*, i.e. sequences of *n* (continuous or discontinuous) items, usually words or morphemes, obtained from a corpus. In most applications, the relevance (i.e. 'MWEhood') of a given ngram is determined using some measure of the strength of the attraction between the items (known as an *association measure*; see Pecina (2005) and Hoang, Kim, and Yan, (2009) for reviews). Additional linguistic and/or statistical filters and thresholds can be used to improve results. The output is a score-ranked list of MWE candidates.

Extraction methods can be classified along several axes: Some methods are designed to extract any type of MWE (Choueka, Klein, and Neuwitz, 1983), while others focus on specific types such as verb-particle constructions (Ramisch et al., 2008) or preposition-noun constructions (Keßelmeier et al., 2009). Some extract only MWEs that contain a specific word/lemma (Kilgarriff and Tugwell, 2001; Cheng et al., 2009), while others extract MWEs without regard to their lexical content (Banerjee and Pedersen, 2003). Another basic parameter is whether or not a given method can deal with discontinuity, i.e. the interruption of a MWEs elements by additional material. Most methods only deal with continuous MWEs (Aires, Lopes, and Silva, 2008), but some deal with both continuous and discontinuous ones (da Silva et al., 1999).

Most extraction algorithms combine statistical methods with linguistic knowledge, which can be integrated into the system in one or more pre- or post-processing steps. This can take several forms such as POS-tagging (Justeson and Katz, 1995; Lossio-Ventura et al., 2014), lemmatization (Daille, 1994; Evert and Krenn, 2001), morphological analysis (Al-Haj and Wintner, 2010; Kumova-Metin and Karaoğlan, 2010), syntactic parsing (Smadja, 1993; Uhrig, Evert, and Proisl, 2018), stop lists (Frantzi and Ananiadou, 1999; Banerjee and Pedersen, 2003), synsets (Pearce, 2001), morphosyntactic patterns (Ramisch, Villavicencio, and Boitet, 2010; Passaro and Lenci, 2016), and semantic tags (Piao et al., 2003; Dunn, 2017). Combining statistical methods with linguistic knowledge involves a trade-off: Methods that use linguistic knowledge may perform better (Wermter and Hahn, 2006, p. 791), but are more language-dependent; while methods that do not use linguistic knowledge are more language-independent, but might have more limited performance.

There are at least four persistent challenges MWE extraction systems faced in their more than fifty-year history. The first is that, although MWEs frequently are longer than two words, virtually all association measures used in MWE extraction are designed to only extract bigrams, i.e. sequences of two items (Wahl and Gries, 2020, p. 88). Several techniques have been proposed to generalize association measures to ngrams longer than two (da Silva et al., 1999; van de Cruys, 2011; Dunn, 2018).

A second challenge is that extraction methods do not behave identically at different frequency ranges (Evert and Krenn, 2001, Section 4.3). For example, the association measure *pointwise mutual information* is known to produce extremely high association scores for low-frequency MWEs, while *t-score* does the same for high-frequency MWEs (Gries, 2010, p. 14). This is a problem even if one tries to use the appropriate measure for the appropriate frequency range. First, it is not easy to accurately describe how a given association measure behaves at different frequencies. Second, determining the exact point where one measure stops being useful and another measure would perform better requires experimentation, and is therefore prone to error. Reduced or zero sensitivity to frequency is a desirable property for an extraction method.

A third problem is that most extraction methods require the setting of one or more parameters for optimum performance. This is problematic because setting a parameter accurately requires experimentation, which is prone to error and introduces the risk of data overfitting. Moreover, the correct value of a parameter depends on various factors such as the language and size of the corpus, the association measures used for extraction, and the type(s) of MWE being extracted (da Silva et al., 1999).

The fourth persistent challenge has been variously referred to as *nested terms* (Frantzi, Ananiadou, and Mima, 2000, p. 117), *overlapping chains* (Mason, 2006, p. 155) and *included components* (O'Donnel, 2011, p. 166). Consider the expression *strawberry ice cream*. Any sentence that contains this trigram also contains the two bigrams *strawberry ice* and *ice cream*. A method that extracts *strawberry ice cream* as a valid MWE because its frequency is high enough would tend to extract the two bigrams as well, since their frequencies will, by definition, be at least as high as that of the original trigram. The problem is that one of the bigrams (*ice cream*) is a valid MWE, while the other (*strawberry ice*) is not, and a purely frequency-based extraction method has no mechanisms to make the correct decision. Several methods have been proposed to deal with this problem (Kita et al., 1994, p. 25; Ren et al., 2009, p. 49; Wei and Li, 2013, p. 519).

3. Proposed Algorithm[1]

3.1 General Characteristics

This paper proposes an algorithm for extracting continuous, i.e. uninterrupted MWEs from a corpus. The algorithm relies on the concept of co-selection in line with Sinclair's (1987) *idiom principle*, according to which "speakers and writers co-select the words they speak and write in order to produce units of meaning, even though the words might appear to be analysable into segments" (quoted in Cheng et al., 2009, p. 239). Since co-selection is a cognitive phenomenon that cannot be observed directly, the algorithm uses textual co-occurrence as a proxy. Therefore, as is the case with other statistical extraction techniques, the results are valid only to the extent this approximation is valid.

The main idea behind the algorithm is to detect the *frequency anomalies* that occur at the starting and ending points of a MWE, which, for purposes of this paper, is defined as a *recurring sequence of linguistic units, i.e. words and/or morphemes*. The algorithm detects these

[1] A Python implementation of the proposed algorithm is available at https://github.com/melanuria/mwe_extractor.

anomalies by manipulating several matrices of ngram frequencies.

The proposed algorithm is *node-based*, i.e. extracts MWEs that contain the item specified by the user, using a fixed window-size around the node. It uses a *candidate generation and ranking* approach, where the input is a set of concordances containing the node, and the output is a score-ranked list of MWE candidates. It is *knowledge-poor*, i.e. does not require linguistic knowledge, except as may be necessary for segmenting the raw input into words or morphemes (Section 3.2). According to the experiment in Section 4, the algorithm seems to be *language-independent*, at least to some extent. Finally, it is *computationally efficient*, with a time complexity of *O(n)*.

3.2 From Concordances to Ngrams

The raw input consists of *N* concordance lines that contain the node specified by the user. Although the node is usually a simplex content word, also bound morphemes, complex word-forms and even multiple word-forms can be used as node. The user also specifies two window sizes, W_L and W_R, for the left and right context of the node, respectively.[2] A pre-processor then converts each of the *N* concordance lines into a sequence of W_L+1+W_R elements (e.g. a 7-gram with the node in the middle, if window size is three on both sides).

The next step is to identify sentence boundaries and punctuation marks, which are treated as *boundary tokens* that MWEs cannot cross. All boundary tokens and any other tokens that are farther away from the node are replaced by the dummy string ###. Finally, position prefixes are added to all tokens, where *Ln* and *Rn* represent the nth token in the left and right contexts, respectively, and *KW* represents the node. Table 1 shows three raw concordance lines and ngrams for English, for a window size of three on both sides.[3]

Concordance1: *and global warming at the same time provide alternative livelihood for the hill indigenous people.* Ngram1 = {L3_at, L2_the, L1_same, KW_time, R1_provide, R2_alternative, R3_livelihood}
Concordance2: *the vehicles will drive ahead and have our camp set up by the time you arrive.* Ngram2 = {L3_up, L2_by, L1_the, KW_time, R1_you, R2_arrive, R3_###}
Concordance3: *profiles the director and looks at his life and work, including time spent with son noel.* Ngram3 = {L3_###, L2_###, L1_including, KW_time, R1_spent, R2_with, R3_son}

Table 1: Raw data and ngrams for W_L=3 and W_R=3

An important question arises at this point: What is the proper unit of analysis for the MWE extraction task, i.e. what should individual ngram elements consist of? Using word-forms may be appropriate for an analytic language like English, because, compared to a less analytic language, an average English lemma has fewer word-forms grouped under it. Consider the light-verb construction *have a hard time*, which has four variants: *has/had/have/having a hard time*. An obvious solution would be to group these word-forms under the lemma HAVE, which would allow us to abstract away from the syntactically motivated surface variation, and represent the MWE as HAVE *a hard time*.

Although lemmatisation is a viable option, the cost of *not* lemmatising is not prohibitively high in English. In the absence of lemmatisation, the total frequency of the construction is divided among the four versions, resulting in some data sparsity, which makes it somewhat harder to extract the construction, and also causing some fragmentation, which means that the candidate list contains four separate entries for the four versions (assuming the algorithm manages to extract them all).

An agglutinating language like Turkish presents a radically different picture. Consider the N-V collocation *-e zaman ayır,* -DAT time spare, ‘to spare time for something’.[4] This construction requires the object to carry a dative marker, which means that, every time the construction is used with a different noun, a different, complex word-form occurs at position L1: *aileme*, family-P1S-DAT, ‘to my family’; *ailelerinize*, family-PL-P2P-DAT, ‘to your families’; *uykuya*, sleep-DAT, ‘to sleep’, etc. Moreover, like many other Turkish verbs, *ayır-* has several thousand different realizations[5], depending on the sequence of suffixes attached to it: *ayırdık*, spare-PAST-1P, ‘we spared’; *ayıramıyorum*, spare-ABIL-NEG-PRES-1S, ‘I cannot spare’; *ayırabilirler*, spare-ABIL-AOR-3P, ‘they can spare’, etc. This means that, when word-forms are used as units, the total frequency of *-e zaman ayır-* is divided among thousands of different word-form trigrams, resulting in extreme data sparsity, which makes it difficult, if not impossible, to extract the construction. Also the fragmentation problem is exacerbated by several orders of magnitude compared to English, meaning that the candidate list contains a very large number of different entries that instantiate the same construction, once again assuming the algorithm manages to extract them. Similar problems caused by the morphology of Turkish have been discussed by several authors in information extraction contexts (Tür, Hakkani-Tür, and Oflazer (2003); Yeniterzi (2011); Eryiğit et al. (2015, pp. 71-72).

In view of the above, it seems appropriate to use word-forms as ngram elements for English data, and individual

[2] A typical setting would be ±5 (see Smadja, 1993, p. 151; Martin, 1983, quoted in Smadja, 1989, p. 6).

[3] Examples are in Turkish and English since the algorithm has been tested on these two languages (Section 4). All examples are based on data obtained from the *trTenTen12* and *enTenTen20* corpora available at sketchengine.co.uk.

[4] The notation used to describe Turkish morphology cannot be covered here in any depth. The following list of glosses are intended to assist the interpretation of the examples in this paper:
Case markers: **ACC** (accusative), **DAT** (dative), **LOC** (locative), **ABL** (ablative), **GEN** (genitive)
Possessive markers on nouns: **P1S**, **P2S**, **P3S**, **P1P**, **P2P**, **P3P**
Plural marker: **PL**
Negation marker: **NEG**
Compound marker: **CM** (identical to **P3S** in form)
Tense/aspect/modality markers: **ABIL** (abilitative), **AOR** (aorist), **CAUS** (causative), **COND** (conditional), **DES** (desiderative), **EVID** (evidential), **FUT** (future), **IMP** (imperative), **NEC** (necessitative), **OPT** (optative), **PAST** (past), **PRES** (present)
Relativizers: **OBJREL** (object), **SUBJREL** (subject)
Person markers on verbs: **1S**, **2S**, **3S**, **1P**, **2P**, **3P**

[5] Although the exact figure is difficult to calculate, a quick corpus query suggests that the trTenTen12 corpus at sketchengine.co.uk contains more than 2,000 unique word-forms (types) based on this verb.

morphemes for Turkish data.[6] To achieve this, Turkish concordance lines have been processed by the morphological analyser described by Çöltekin (2010), which generates *all* possible analyses for each word-form. And this brings us to the problem of *morphological ambiguity*. Consider the following sentence:

Ürünü istediği zaman alabileceğini bilen müşteri, alımı erteler.
'Knowing that he/she can purchase the product any time he/she wants, the customer postpones the purchase.'

For the node *zaman*, 'time', and a window size of five on both sides, the word-forms *ürünü*, *istediği* and *alabileceğini* are morphologically ambiguous, each having two possible morphological analyses. This results in eight possible morpheme sequences (ambiguities underlined):

ürün-ACC *iste*-OBJREL-ACC *zaman al*-ABIL-FUT-CM-ACC
ürün-CM *iste*-OBJREL-ACC *zaman al*-ABIL-FUT-CM-ACC
ürün-ACC *iste*-OBJREL-CM *zaman al*-ABIL-FUT-CM-ACC
ürün-CM *iste*-OBJREL-CM *zaman al*-ABIL-FUT-CM-ACC
ürün-ACC *iste*-OBJREL-ACC *zaman al*-ABIL-FUT-P2S-ACC
ürün-CM *iste*-OBJREL-ACC *zaman al*-ABIL-FUT-P2S-ACC
ürün-ACC *iste*-OBJREL-CM *zaman al*-ABIL-FUT-P2S-ACC
ürün-CM *iste*-OBJREL-CM *zaman al*-ABIL-FUT-P2S-ACC

To be able to use individual morphemes rather than word-forms as their unit of analysis, several studies on information extraction in Turkish (Küçük and Yazıcı, 2009; Kumova-Metin and Karaoğlan, 2010; Yeniterzi, 2011; Şeker and Eryiğit, 2012; Kazkılınç, 2013, Güngör, Güngör, and Üsküdarlı, 2019) have resorted to *morphological disambiguation* (*i.e.* a mechanism that selects one of the available morphological analyses as the "correct", or at least the most probable, one). But this is dangerous in a MWE extraction setting because morphological disambiguation in agglutinating languages is not a trivial task and its performance relies, among several other factors, on the proper handling of MWEs. In other words, using a morphological disambiguator in a MWE extraction algorithm amounts to using the output of a task to perform another task when the outcome of the former depends on the latter. This is why the proposed algorithm refrains from disambiguating the morphological analyses. Instead, whenever there are more than *n* possible analyses, it randomly chooses *n* of them. This is an obviously more inferior but more cautious approach.

In an experimental step to deal with morphological variability in Turkish, possessive markers on nouns are replaced by the 'super-tag' POSS. To draw a parallel to English, this allows the system to treat, say, *for the first time in my/your/his/her/its/our/their life/lives* as instances of the abstract MWE *for the first time in one's life*.

The last step for both English and Turkish is to pre-calculate the following global frequencies:

- Position-specific frequency of every token (e.g. frequency of *spent* at position R_1); and
- position-specific frequency of each of the $(W_L+1)\times(W_R+1)$ uninterrupted, node-containing sub-sequences of the *N* concordance lines (e.g. frequencies of *same time*, *same time provide*, etc.)

3.3 Observed Frequencies

The co-selection matrix of observed frequencies, *O*, is a W_L+1 by W_R+1 matrix that stores the observed ngram frequencies the algorithm uses to extract MWEs:

$$\mathrm{O} = \begin{bmatrix} f(KW) & f(KW \cdots R_1) & f(KW \cdots R_2) & f(KW \cdots R_3) \\ f(L_1 \cdots KW) & f(L_1 \cdots R_1) & f(L_1 \cdots R_2) & f(L_1 \cdots R_3) \\ f(L_2 \cdots KW) & f(L_2 \cdots R_1) & f(L_2 \cdots R_2) & f(L_2 \cdots R_3) \\ f(L_3 \cdots KW) & f(L_3 \cdots R_1) & f(L_3 \cdots R_2) & f(L_3 \cdots R_3) \end{bmatrix}$$

Row and column indices correspond to the left and right context of the node, respectively. Each matrix element stores the observed frequency of an uninterrupted sub-sequence that starts at the token represented by the row-index and terminates at the token represented by the column-index. For instance, matrix element $O_{4,3}$ for Ngram1 in Table 1 stores the observed frequency of the 6-gram that starts at L_3 and ends at R_2 (shorthand notation $L_3...R_2$), i.e. the sub-sequence *at the same time provide alternative*. In other words, each matrix element shows how many times the corresponding sub-sequence of an individual ngram occurs in the entire input.

The topological organization of the matrix is such that moving from a given matrix element to the element on the right represents adding a new token to the right of the original sequence, and moving to the element below represents adding a new token to its left. The top-left element, $O_{1,1}$, which represents the bare node, is the starting point, and the sub-sequences get incrementally longer as one moves from there to the bottom-right element, which represents the longest sequence determined by W_L and W_R.

Critically, each of the *N* concordance lines included in the analysis has its own co-selection matrix. The co-selection matrix is a *local* artefact that allows the algorithm to select the best-performing sub-sequence(s) of a single ngram, using *global* frequency values obtained from the entire input data.

3.4 Adjusting Observed Frequencies

The next step is to deal with the *nesting problem* discussed in Section 2 by adjusting the co-selection matrix of observed frequencies. In mathematical terms, the problem is that every sub-sequence $L_m...R_n$ contains ((m+1) × (n+1)) - 1 shorter sub-sequences, which means that, whenever the frequency of $L_m...R_n$ is incremented, the frequencies of each of those shorter sub-sequences are incremented as well. To prevent this repetitive counting, matrix *O* is processed element-by-element, starting at the bottom-right corner and proceeding diagonally to the shorter sub-sequences, until the top-left corner is reached. At every step, the frequency of the sub-sequence being processed is deducted from the frequencies of all shorter sub-sequences. The end result is *O'*, the *adjusted co-selection matrix of observed frequencies*.

Below is an example for Ngram1 in Table 1:

$$O'_{Ngram1} = \begin{bmatrix} 47880 & 3 & 0 & 0 \\ 42 & 0 & 0 & 0 \\ 195 & 0 & 0 & 0 \\ 1754 & 0 & 0 & 0 \end{bmatrix}$$

[6] This is not a dichotomy but a continuum. It seems safe to assume that the more synthetic a language is, the more it would benefit from a morpheme-based treatment.

3.5 Expected Frequencies and Aggregate Matrix

3.5.1 Definitions

The proposed algorithm works by comparing O' to either the co-selection matrix of expected frequencies (E), or to the aggregate matrix (A). The following definitions are needed to describe these two methods:

Definition 1: The probability of observing a given token at a given position is approximated by dividing the number of times that token occurs at that position by the number of ngrams included in the analysis:

$$p(R2\_arrive) = \frac{f(R2\_arrive)}{N}$$

Definition 2: The probability of *not* observing a given token at a given position is approximated by taking the complement of the probability of observing that token in that position:

$$p(R2\_arrive') = 1 - \frac{f(R2\_arrive)}{N}$$

Definition 3: The expected probability of observing a sequence $L_m...R_n$ is approximated by multiplying the probabilities of observing each token in the sequence, the probability of *not* observing L_{m+1}, and the probability of *not* observing R_{n+1}.[7] For example, in relation to Ngram1 in Table 1:

$$p(L_2 \cdots R_1)_{Ngram1} = p(L2\_the) \times p(L1\_same) \times$$

$$p(R1\_provide) \times p(L3\_at') \times p(R2\_alternative')$$ [8]

Definition 4: The co-selection matrix of expected frequencies (E) is calculated by applying Definition 3 to each sub-sequence in O', and multiplying the resulting matrix by the scalar N, to convert expected probabilities to expected frequencies:

$$E = N \begin{bmatrix} p(KW) & p(KW \cdots R_1) & p(KW \cdots R_2) & p(KW \cdots R_3) \\ p(L_1 \cdots KW) & p(L_1 \cdots R_1) & p(L_1 \cdots R_2) & p(L_1 \cdots R_3) \\ p(L_2 \cdots KW) & p(L_2 \cdots R_1) & p(L_2 \cdots R_2) & p(L_2 \cdots R_3) \\ p(L_3 \cdots KW) & p(L_3 \cdots R_1) & p(L_3 \cdots R_2) & p(L_3 \cdots R_3) \end{bmatrix}$$

Definition 5: The aggregate matrix A is equal to the matrix-sum of the N adjusted co-selection matrices of observed frequencies:

$$A = \sum_{i=1}^{N} O'_i$$

3.5.2 Using the Co-selection Matrix of Expected Frequencies to Detect Anomalies

The co-selection matrix of expected frequencies of a given ngram (E) contains the expected frequencies of each sub-sequence in O'. Just as every individual ngram has its own O', every individual ngram has its own E. The expected frequencies matrix provides a baseline for detecting anomalies in an O' matrix:

$$E_{Ngram1} = \begin{bmatrix} 49598 & 209 & 0 & 0 \\ 65 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

According to E_{Ngram1}, the expected frequency of $L_3...KW_{Ngram1}$ (the sub-sequence *at the same time*) is zero. Since the corresponding frequency in O'_{Ngram1} (f=1754, Section 3.4) is significantly higher than zero, the sequence *at the same time* has a high probability of being a MWE.

3.5.3 Using the Aggregate Matrix to Detect Anomalies

The aggregate matrix A shows how the total probability mass is distributed among matrix elements *in the aggregate*. There is only one aggregate matrix for every node word, and the sum of its elements is always equal to 1.0. Just like E, A provides a baseline for detecting anomalies in individual O matrices.

The aggregate matrix for *time*:[9]

$$A_{time} = \begin{bmatrix} 0.9479130 & 0.0158255 & 0.0003455 & 0.0000100 & 0.0000005 & 0.0000002 \\ 0.0275413 & 0.0008389 & 0.0001221 & 0.0000020 & 0.0000003 & 0.0000008 \\ 0.0048604 & 0.0002991 & 0.0000333 & 0.0000010 & 0.0000002 & 0.0000012 \\ 0.0020924 & 0.0000586 & 0.0000026 & 0.0000003 & 0.0000002 & 0.0000009 \\ 0.0000314 & 0.0000013 & 0.0000002 & 0.0000001 & 0.0000001 & 0.0000009 \\ 0.0000070 & 0.0000009 & 0.0000010 & 0.0000009 & 0.0000008 & 0.0000047 \end{bmatrix}$$

According to this, an average O' matrix for the node *time* is expected to have 2.75% of its total frequency in the matrix element $O'_{2,1}$. If an individual O' matrix has significantly more than 2.75% of its total frequency in $O'_{2,1}$, this would indicate that the sub-sequence represented by that element ($L_1...KW$) has a higher-than-average probability of being a MWE.

3.6 Calculating Scores

A distinctive feature of the proposed algorithm is that a separate O and a separate E, and consequently a separate score matrix S is generated for each of the N items in the input data. This allows the algorithm to *locally* select only those sub-sequences that have the highest probability of being a MWE, thus preventing the remaining sub-sequences from 'contaminating' the statistics. Considering that most existing methods indiscriminately generate all possible sub-sequences of a given ngram, the proposed method ensures a dramatic[10] reduction in the amount of data that will have to be considered during score-aggregation and ranking.

[7] When m=W_L and/or n=W_R (i.e. along the bottom and right edges of the matrix), the probabilities of not observing L_{m+1} and R_{n+1} are undefined, and are thus assumed to be 1.0.

[8] *p(KW)* can be omitted because it is by definition equal to 1.0 (all ngrams contain the node *KW* in the middle).

[9] Unlike the earlier examples, this example uses a window size of five on both sides. For ease of presentation, the matrix has been normalized by dividing it by the sum of its elements.

[10] A (5+1)×(5+1)=36-fold reduction for a typical window-size of 5 on both sides, assuming the algorithm selects a single top-performing candidate from each score matrix.

As mentioned in Section 3.5.1, S is calculated by comparing O' to either A or E. In the former case, S is simply equal to O'/A. In the latter case:

$$S = \frac{\log_2(O' + 1)}{\log_2(E + a)}$$

where a is a constant correction factor to avoid logarithms of zero (and one of the parameters in the experiment in Section 4).

A possible modification to the score matrix is *length adjustment*, where every element of S is divided by the length of the sub-sequence represented by that element. Length adjustment is another parameter in the experiment described in Section 4.

3.7 Selecting Candidates

Having obtained N score matrices for the N concordance lines, the next step is to select the best MWE candidate(s) that each concordance line will forward to the score aggregation and ranking stage. Two parameters relevant at this point are c, the number of candidates to be selected from each score matrix, and t, the minimum score required for being selected. In formal terms, the set of candidates consists of the c ngrams whose score in S is equal to or greater than t. If c=3 and t=1.5, for instance, three sub-sequences with the highest scores will be selected, and those with a score of 1.5 or higher will be forwarded to the score aggregation stage.

3.8 Score Aggregation and Candidate-Ranking

The next step is to aggregate the scores of the candidates selected in the previous step. Three methods will be tested for this purpose. In the first method named '*add-one*', the aggregate score of a MWE candidate is incremented by one every time the score-selection algorithm selects it. In the second one named '*add-score*', aggregate score is incremented by the candidate's score in S every time it is selected. In the third one named '*max*', aggregate score is equal to the highest score a candidate obtains in any of the score matrices that select it.

The result of this final step is a score-ranked list of MWE candidates. Top thirty candidates generated by the algorithm for the English word *time* and the Turkish word *zaman*, 'time', are given in Table 2, for N=50,000, and using *Method A* described in Section 4.3.

Rank	English	Turkish
1	*at the same time*	*son zamanlarda*
2	*from time to time*	*her zamanki gibi*
3	*for the first time*	*o zaman*
4	*at the time*	*uzun zamandır*
5	*this time*	*-dıkları zaman*
6	*for a long time*	*kimi zaman*
7	*over time*	*bu zamana kadar*
8	*at that time*	*o zamana kadar*
9	*at this time*	*bir zamanlar*
10	*for the first time in*	*her zamankinden daha*
11	*all the time*	*işte o zaman*
12	*most of the time*	*ne zaman*
13	*a lot of time*	*hiç bir zaman*
14	*at the time of the*	*-e baktığımız zaman*
15	*at a time*	*zaman*
16	*at any time*	*her zaman olduğu gibi*
17	*for some time*	*istediği zaman*
18	*in time*	*-masının zaman*
19	*in real time*	*gerçek zamanlı*
20	*at the time of*	*olduğu zaman*
21	*it is time to*	*her zaman*
22	*of time*	*kısa zaman*
23	*during this time*	*dediği zaman*
24	*at a time when*	*uzun zamandan beri*
25	*every time*	*ne kadar zaman*
26	*of all time*	*ilerleyen zamanlarda*
27	*the time*	*baktığın zaman*
28	*and at the same time*	*o zamandan beri*
29	*for the time being*	*zaman diliminde*
30	*at the right time*	*-mak için zaman*

Table 2. Top-30 candidates for *time* and *zaman*, 'time'

4. Evaluation

4.1 General

The standard approach to evaluating an information extraction system is to report both precision and recall, but this is not a straightforward task in a MWE extraction context. The main problem is that a gold standard against which to compare the results is difficult to define and obtain. One could use an existing resource like a machine-readable dictionary or a wordnet (Schone and Jurafsky, 2001), or a database specifically designed to evaluate MWE extraction systems (Kumova-Metin and Taze, 2017). But such resources are not available for all languages, and their coverage of MWEs is far from complete. Alternatively, one could use what Constant et al. (2017) refer to as *post hoc human judgment*, where each entry in a score-ranked candidate list is manually marked either as a MWE or a non-MWE by one or more experts (p. 853).

The second question is whether to report both precision and recall, or just precision. Most authors have chosen the former alternative (Smadja, 1993; Evert and Krenn, 2001; Eryiğit et al., 2015; Taşçıoğlu and Kumova-Metin, 2021), although several others report only precision (Shimohata, Sugio, and Nagata, 1997; Zhai, 1997; Frantzi, Ananiadou, and Mima, 2000; Dias, 2003). Reporting recall assumes that the researcher has access to the set of all MWEs in a language (or at least the set of all MWEs in the sample used in the study), while reporting precision involves the more reasonable assumption that it is possible to know whether or not a given sequence is a MWE.

This study will refrain from reporting recall. This is because the number of MWEs one finds in a corpus is closely linked to how broadly one chooses to define phraseology. MWE extraction has a relatively short history, and the true extent of the phraseological tendency in human languages is still not sufficiently explored. In other words, we cannot safely assume that we know "the set of all MWEs", or even what it means to know such a thing. It thus seems to be more appropriate to initially adopt a broad definition of phraseology, and then reduce its scope to the extent required by the data.

The 'broad definition of phraseology' adopted in this paper uses the following settings for the six parameters proposed by Gries (2008, p. 4):

i. a MWE may consist of roots or affixes, but must contain at least one lexically specified element;
ii. a MWE must have at least two elements, and cross at least one word boundary (no upper limit to the number of elements);
iii. the observed frequency of a MWE must be higher than its expected frequency;
iv. the elements of a MWE may not be interrupted by other elements (i.e. continuous MWEs only);
v. MWEs may exhibit lexical, syntactic and morphological variability;
vi. a MWE must constitute a semantic unit but does not have to be semantically non-compositional.

The design of the algorithm and the nodes selected already make sure that MWE candidates comply with (i), (ii) and (iii). So, the expert only has to focus on (iv), according to which *have a good time* is a MWE but *have an unexpectedly and unbelievably good time* is not; on (v), according to which *spend quality time* and *spent quality time* are both valid MWEs; and on (vi), according to which *time limit* is a MWE but *time by* is not (semantic unity required), and both *time and again* and *time and date* are MWEs (semantic non-compositionality allowed but not required).

Using the above criteria, the expert marked 1672, 2132 and 1053 sequences as valid MWEs for the three node words selected in Section 4.2, respectively.[11] Although items marked as valid MWEs involve some redundancy (i.e. several variants of the same MWE marked separately), these numbers are still unexpectedly high, suggesting that the phraseological tendency in both English and Turkish is stronger than generally assumed, at least when a broad definition of phraseology is adopted. Existing MWE repositories for Turkish (Eryiğit, İlbay, and Can, 2011; Adalı et al., 2016; Kumova-Metin and Taze, 2017; Berk, Erden, and Güngör, 2018) contain 4,000-30,000 MWEs for the entire language. Thus, they cannot be used as a gold standard in a study that adopts a broad definition of phraseology, where a single word can have around one thousand MWEs.

The third question is how to calculate precision. One option is to report the number of true positives among the top 100 or 200 items on the ranked candidate list. Evert and Krenn (2001) criticize this approach, stating that evaluation results would then be based on a small and arbitrary subset of the candidates, which means that "results achieved by individual measures may very well be due to chance" (p. 2). Instead, they calculate precision at every point of the candidate list, which allows them to plot it as a curve (also see Zhai, 1997, p. 6). The precision curve has been adopted by several authors, and seems to have become a standard in the field (Schone and Jurafsky, 2001; Pecina, 2005; Kumova-Metin, 2016).

A final point is whether or not to use a baseline against which the algorithm's performance can be compared. The *naïve ngram* method is frequently used for this purpose. This consists of generating every possible sub-sequence of every ngram included in the study. The baseline is then created either by calculating the probability of a randomly selected sub-sequence being a MWE (Pecina, 2005), by sorting the sub-sequences in decreasing order of frequency and calculating one or more precision values for some portion of that sorted list (Wermter and Hahn, 2004), or both (Krenn and Evert, 2001). As noted by several researchers (Frantzi, Ananiadou, and Mima, 2000, p. 117; Krenn and Evert, 2001, Section 10; Wermter and Hahn, 2004, Section 4.1), the naïve ngram method performs surprisingly well despite its simplicity. Section 4.3 confirms this finding.

In light of the above discussion, this paper will evaluate the proposed algorithm by reporting precision only (using precision curves based on *post hoc* human judgment), by using the naïve ngram method as a baseline, and by designing an experiment that covers all possible combinations of the algorithm's parameters.

4.2 Experiment Design

The algorithm's performance will be evaluated in an experiment that uses various parameter settings. Throughout the discussion in Section 3, the following emerged as possible parameters:

- Observation matrix O can be used with or without nesting adjustment (Section 3.4);
- score matrix S can be calculated using either expected frequencies matrices (E) or the aggregate matrix (A) (Section 3.5);
- score matrix S can be used with or without length adjustment (Section 3.6);
- the correction factor a (Section 3.6) can have different values (2, 4 and 8 selected for experiment);
- different values can be used for c (Section 3.7) (1, 2 and 3 selected for experiment);
- different values can be used for t (Section 3.7) (0, 0.5, 1 and 2 selected for experiment);
- three methods are available for score aggregation (*add-one*, *add-score*, *max*) (Section 3.8)

Accordingly, there are 2×2×2×3×3×4×3=864 possible parameter combinations. The experiment will run the algorithm once for each of these combinations, and evaluate results.

Since the algorithm will be run with 864 different settings, the resulting unified lists contain a large number of candidates, which makes it impracticable to evaluate more than a few items. In view of this, only three items have been included in the experiment (see Section 5 for a discussion of this choice). Since the aim is to test Turkish and English, and MWE-rich and MWE-poor items, the selection consists of the words *time* (expected to be MWE-rich), *zaman*, 'time' (expected to be MWE-rich), and *literatür*, 'academic literature' (expected to be MWE-poor).

The MWE candidate lists for these items were manually annotated by the author (6,190 candidates for *time*, 17,236 candidates for *zaman*, and 10,305 candidates for *literatür*). To minimize bias, the 864 candidate lists generated by the algorithm and the candidate list generated by the naïve ngram method were combined, and the lines randomized.

[11] The manually marked gold-standard files for the three node words are available at https://github.com/melanuria/mwe_extractor/tree/main/data.

This ensured that the annotator had no way of knowing if a given candidate was generated by the algorithm or by the naïve ngram method. Even if the annotator somehow guesses that a candidate was generated by the algorithm, he/she cannot know which of the 864 versions generated it.

4.3 Experiment Results

The nodes *time* and *zaman* were included in the experiment because they refer to the same, very basic, concept in English and Turkish, and are thus expected to be a part of a large number of MWEs, while *literatür* was included for its highly-specialized meaning, expected to result in fewer MWEs. As expected, the two cases had two different best-performing parameter combinations (Table 3), and different precision profiles (Figures 1-4).

Parameter	*Method A*	*Method B*
nesting adjustment	yes	no
comparison method	E matrix	A matrix
length adjustment	none	none
correction factor (*a*)	2	2
number of candidates (*c*)	1	2
score threshold (*t*)	0	0
score aggregation method	add-one	add-one

Table 3. Two best-performing parameter combinations

The combination that performed best for *time* and *zaman* will be named *Method A*, and the one that performed best for *literatür Method B*. Figures 1-3 show the precision curves Method A generated for the three nodes, in each case for the top-1000 candidates. Figure 4 shows the precision curves Method B generated for *literatür*, again for the top-1000 candidates. A dashed line shows the precision of the naïve ngram method, a solid line the precision of the best parameter combination, and thin grey lines the precisions of the remaining 863 combinations.

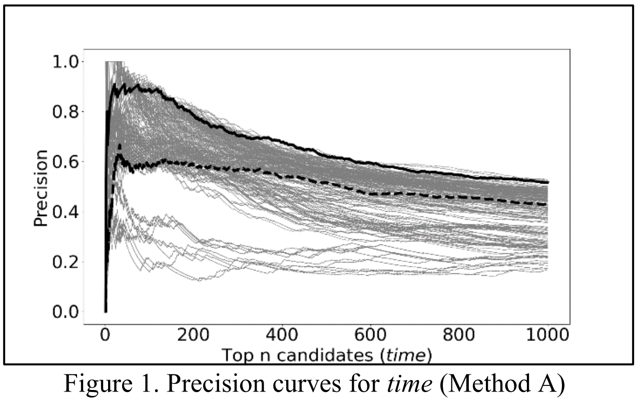

Figure 1. Precision curves for *time* (Method A)

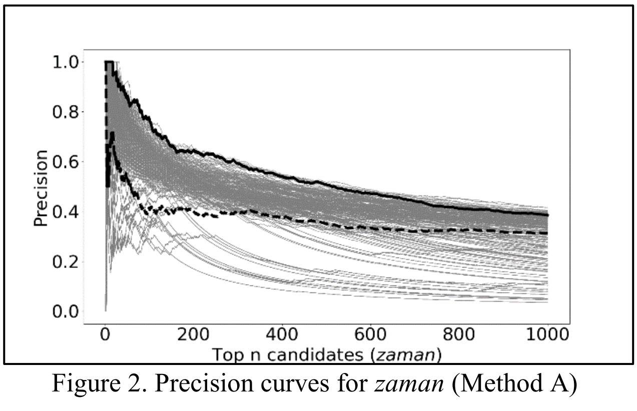

Figure 2. Precision curves for *zaman* (Method A)

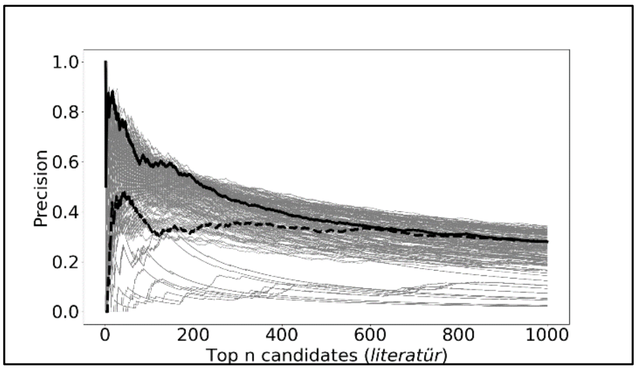

Figure 3. Precision curves for *literatür* (Method A)

The performance of the naïve ngram method confirms findings in the literature. Despite its extreme simplicity, it provides 50-60% precision for the first few hundred items, and 30-40% at n=1000, regardless of the node-word used.

The proposed algorithm gives promising results, especially for the top few hundred items of the candidate lists. For all three nodes, Method A generates top-50 precision values between 0.71 and 0.88, top-100 precision values between 0.60 and 0.88, and top-200 precision values between 0.54 and 0.78. Thus, in applications where a minimum precision of around 0.70 is acceptable and only the most prominent 50 or so MWEs of a word are required, Method A can be used without post-processing. In applications that require larger and more precise MWE lists, the same method can be used to obtain more than 100 MWEs per word, with the manual effort of reviewing the top 150-200 candidates. When the algorithm is used to process, say, the most frequent 20,000 words of a language, the resulting MWE lexicon would probably contain more than one million entries, even after accounting for redundancies.

For the MWE-rich items *time* and *zaman*, Method A consistently performs 20-35 percentage points above the baseline up to n=200, and retains a 10-point lead even at n=1000. For the MWE-poor *literatür*, however, Method A falls towards the baseline more quickly, finally converging with it at around n=600 (Figure 3). In contrast, Method B performs consistently above baseline for this node word, even at n=1000 (Figure 4).

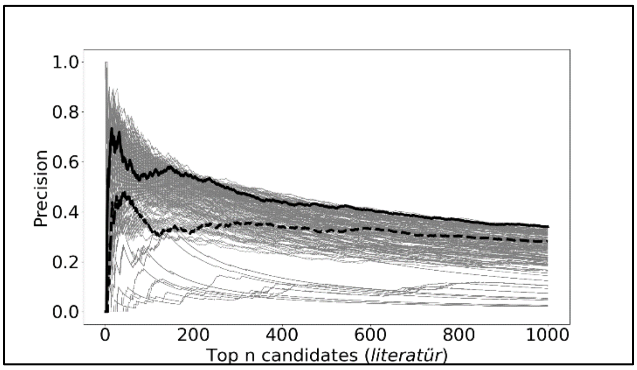

Figure 4. Precision curves for *literatür* (Method B)

Although additional evaluation data is required to reach statistically meaningful conclusions, existing results suggest that Method A provides an efficient method for automatically extracting the phraseology of relatively more frequent and general-purpose words, and/or extracting the most prominent MWEs of each word, while Method B can be used to extract the phraseology of relatively less frequent words with a more specialized meaning, and/or to obtain higher precision at the bottom of the candidate lists.

5. Conclusion

This paper proposed and evaluated an algorithm for automatically extracting MWEs from a corpus. Initial results show that it works equally well for two typologically different languages, English and Turkish.

The algorithm uses a *co-selection matrix* that gradually adds elements to the left and right contexts of a starting element (the node), and works by detecting the frequency anomalies that occur at the starting and ending points of a MWE. It is in this regard conceptually similar to a family of existing algorithms including the *neighbour-selectivity index* algorithm by Choueka et al. (1983), the *Xtract* algorithm by Smadja (1993), and the *LocalMaxs* algorithm by da Silva and Lopes (1999). The most important difference between the proposed algorithm and these earlier algorithms is that the proposed algorithm is node-based, knowledge-poor and computationally efficient. Another important difference is that it can be used to for both *extraction* and *identification*, the latter being "the process of automatically annotating MWE tokens in running text by associating them with known MWE types" (Constant et al., 2017). This is because the algorithm generates matrices for individual input sequences, and can thus determine the top-performing sub-sequences of any sequence entered by the user.

The algorithm has certain properties that address some of the recurring issues in MWE extraction (Section 2). First, it avoids using association measures, which are generally limited to detecting the association between two items. This means that the algorithm can extract sequences of arbitrary length, as long as length does not exceed window size. Second, it solves the frequency sensitivity problem to a certain extent in that the final ranking strictly follows the overall frequency order of the relevant candidates, which means that low-frequency items are not disproportionately pushed to the top of the list, and vice versa. Third, it avoids the nesting problem by applying the adjustment described in Section 3.4, and also by selecting a user-defined number of top-performing sub-sequences from a given ngram and ignoring all remaining sub-sequences. Fourth, it achieves a relatively high precision although it does not require morpho-syntactic patterns or other linguistics filters. In this sense, the algorithm seems to refute Frantzi and Ananiadou (1999), who claim that "the statistical information that is available, without any linguistic filtering, is not enough to produce useful results" (p. 147), and also Wermter and Hahn (2006), who claim that "purely statistics-based measures reveal virtually no difference compared with frequency of occurrence counts, while linguistically more informed metrics do reveal such a marked difference" (p. 785).

The present version of the algorithm also has certain limitations. First, it does not deal with certain types of MWE *variability*, a main challenge in MWE processing (Constant et al., 2017, p. 848). *Morphosyntactic variability* has already been dealt with to some extent (Section 3.2). In contrast, it does not seem easy to generalize the algorithm to deal with *positional variability*, where the order of the elements changes (e.g. *agreement signed by X* vs. *X signed an agreement*).

Second, the algorithm cannot extract discontinuous MWEs, another main challenge in MWE processing (Constant et al., 2017, p. 848). Future work could focus on this limitation as well. One promising avenue is to extend the algorithm to *phrase frames* ("ngrams with one variable slot") and PoS-grams ("a string of part-of-speech categories") (Stubbs, 2007, pp. 90-1). This might be achieved by manipulating the co-selection matrices such that they contain a mixture of lexical items and POS tags, and by treating certain matrix rows and/or columns as *slots* that accept only certain lexical items that have the same part-of-speech or belong to the same semantic class, or only certain affixes that belong to the same paradigm. Another idea would be to combine the method with knowledge-rich pre- or post-processing steps to improve precision.

Third, the algorithm has been evaluated on three words only, and this limits the validity of the results reported in Section 4. The total annotator time available could be allocated to increase (a) the number of experiment parameters tested, (b) the number of words tested, or (c) the number of candidates per word. This being an initial report on the proposed algorithm, it seemed more reasonable to maximize (a) and (c) at the expense of (b), i.e. to test a few candidate lists thoroughly (n=1000) for all possible combinations of the algorithm's parameters. Future work should focus on increasing (b) without compromising (a) or (c), and also increasing the number of reviewers and adding inter-judge agreement to the picture.

The original aim of this study was to design an algorithm to extract Turkish MWEs of arbitrary length. This was partially in response to Biber (2009), who stated that research was required to document sequences that are longer than two words, and asked "how are formulaic expressions realized in other languages; for example, in morphology-rich languages like Finnish or Turkish?" Biber thinks that "different linguistic devices will be required to realize formulaic expressions in these languages" and that "it is not even clear that formulaic language will be equally important in all languages" (p. 301).

The proposed algorithm focuses on three of the more superficial and quantifiable properties of MWEs: (a) A MWE crosses at least one word boundary; (b) a MWE is a sequence of co-selected linguistic elements that function as a single semantic unit; and (c) the elements of a MWE co-occur more frequently than expected. The fact that such a linguistically impoverished algorithm works equally well for English and Turkish suggests that the essential characteristics of the phraseologies of typologically different languages might not be as divergent as Biber thought. Moreover, the fact that 50,000 concordance lines can produce more than one thousand MWE types containing the same word suggests that formulaic language might very well be "equally important in all languages", and probably more important than generally assumed.

6. Bibliographical References

Adalı, K., Dinç, T., Gökırmak, M., and Eryiğit, G. (2016). Comprehensive annotation of multiword expressions for Turkish. *Proceedings of TurCLing*, 60–66.

Aires, J., Lopes, G., and Silva, J. F. (2008). Efficient multi-word expressions extractor using suffix arrays and related structures. *Proceedings of the 2nd ACM Workshop on Improving Non English Web Searching*, 1–8.

Al-Haj, H., and Wintner, S. (2010). Identifying multi-word expressions by leveraging morphological and syntactic idiosyncrasy. *Proceedings of the 23rd International conference on Computational Linguistics*, 10–18.

Baldwin, T., and Kim, S. N. (2010). Multiword Expressions. In N. Indurkhya and F. J. Damerau (Eds.), *Handbook of Natural Language Processing* (Second Edition, pp. 267–292). CRC Press.

Banerjee, S., and Pedersen, T. (2003). The design, implementation, and use of the Ngram statistics package. *Lecture Notes in Computer Science*, 2588, 370–381.

Berk, G., Erden, B., and Güngör, T. (2018). Turkish verbal multiword expressions corpus. *26th Signal Processing and Communications Applications Conference (SIU)*, 1–4.

Biber, D. (2009). A corpus-driven approach to formulaic language in English: Multi-word patterns in speech and writing. *International Journal of Corpus Linguistics*, 14(3), 275–311.

Calzolari, N., Fillmore, C. J., Grishman, R., Ide, N., Lenci, A., MacLeod, C., and Zampolli, A. (2002). Towards Best Practice for Multiword Expressions in Computational Lexicons. *LREC 2002*.

Cheng, W., Greaves, C., Sinclair, J. M., and Warren, M. (2009). Uncovering the extent of the phraseological tendency: Towards a systematic analysis of concgrams. *Applied Linguistics*, 30(2), 236–252.

Choueka, Y., Klein, S. T., and Neuwitz, E. (1983). Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. *Journal for Literary and Linguistic Computing*, 4(1), 34–38.

Church, K. W., Gale, W. A., Hanks, P., and Hindle, D. (1991). Using Statistics in Lexical Analysis. *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, January 1991, 115–164.

Constant, M., Eryiğit, G., Monti, J., Plas, L. van der, Ramisch, C., Rosner, M., and Todirascu, A. (2017). Multiword Expression Processing: A Survey. *Computational Linguistics*, 43(4), 837–892.

Croft, W., and Cruse, D. A. (2004). *Cognitive Linguistics*.

Çöltekin, Ç. (2010). A freely available morphological analyzer for Turkish. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).*

Da Silva, J. F., and Lopes, G. P. (1999). A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. *Sixth Meeting on Mathematics of Language*, 369–381.

Da Silva, J. F., Dias, G., Guilloré, S., and Lopes, J. G. P. (1999). Using LocalMaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. *Lecture Notes in Computer Science*, 1695, 113–132.

Daille, B. (1994). Study and implementation of combined techniques for automatic extraction of terminology. *The balancing act: Combining symbolic and statistical approaches to language*.

De Cruys, T. (2011). Two multivariate generalizations of pointwise mutual information. *Proceedings of the Workshop on Distributional Semantics and Compositionality*, 16–20.

Dias, G. (2003). Multiword unit hybrid extraction. *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, 41–48.

Dunn, J. (2017). Computational learning of construction grammars. *Language and Cognition*, 9(2), 254–292.

Dunn, J. (2018). Multi-unit association measures : Moving beyond pairs of words. *International Journal of Corpus Linguistics*, 23(2), 183–215.

Erman, B., and Warren, B. (2000). The idiom principle and the open choice principle. *Text*, 20(1), 29–62.

Eryiğit, G., Adalı, K., Torunoğlu-Selamet, D., Sulubacak, U., and Pamay, T. (2015). Annotation and extraction of multiword expressions in Turkish treebanks. *Proceedings of the 11th Workshop on Multiword Expressions*, 70–76.

Eryiğit, G., İlbay, T., and Can, O. A. (2011). Multiword Expressions in Statistical Dependency Parsing. *Proceedings of the 2nd Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL 2011)*, 45–55.

Evert, S., and Krenn, B. (2001). Methods for the qualitative evaluation of lexical association measures. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 188–195.

Frantzi, K. T., and Ananiadou, S. (1999). The C-value/NC-value domain-independent method for multi-word term extraction. *Journal of Natural Language Processing*, 6(3), 145–179.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2), 115–130.

Grant, L., and Bauer, L. (2004). Criteria for Re-defining Idioms: Are We Barking Up the Wrong Tree? *Applied Linguistics*, 25(1), 38–61.

Gries, S. T. (2008). Phraseology and linguistic theory: A brief survey. In *Phraseology: An interdisciplinary perspective* (pp. 3–25).

Gries, S. T. (2010). Corpus linguistics and theoretical linguistics: A love--hate relationship? Not necessarily…. *International Journal of Corpus Linguistics*, 15(3), 327–343.

Güngör, O., Güngör, T., and Üsküdarlı, S. (2019). The effect of morphology in named entity recognition with sequence tagging. *Natural Language Engineering*, 25(1), 147-169.

Hoang, H. H., Kim, S. N., and Kan, M.-Y. (2009). A Re-examination of Lexical Association Measures. *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation, Applications*, 31–39.

Jackendoff, R. (1997). *The Architecture of the Language Faculty*. MIT Press.

Justeson, J. S., and Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1), 9–27.

Kazkılınç, S. (2012). *Türkçe Metinlerin Etiketlenmesi* [Master's Thesis, Istanbul Technical University].

Keßelmeier, K., Kiss, T., Müller, A., Roch, C., Stadtfeld, T., and Strunk, J. (2009). Mining for Preposition-Noun Constructions in German. *Workshop on Extracting and Using Constructions in Natural Language Processing*.

Kilgarriff, A., and Tugwell, D. (2001). Word sketch: Extraction and display of significant collocations for lexicography.

Kita, K., Kato, Y., Omoto, T., and Yano, Y. (1994). A comparative study of automatic extraction of collocations from corpora: Mutual information vs. cost criteria. *Journal of Natural Language Processing*, 1(1), 21–33.

Kjellmer, G. (1987). Aspects of English collocations. In *Corpus linguistics and beyond* (pp. 133-140). Brill.

Krenn, B., Evert, S., and others. (2001). Can we do better than frequency? A case study on extracting PP-verb collocations. *Proceedings of the ACL Workshop on Collocations*, 39, 46.

Kumova-Metin, S., and Karaoğlan, B. (2010). Collocation extraction in Turkish texts using statistical methods. *International Conference on Natural Language Processing*, 238–249.

Kumova-Metin, S., and Taze, M. (2017). A procedure to build multiword expression data set. *2nd International Conference on Computer and Communication Systems*, 46–49.

Kumova-Metin, S. (2016). Neighbour unpredictability measure in multiword expression extraction. *Comput. Syst. Sci. Eng*., 31, 209–221.

Küçük, D., and Yazıcı, A. (2009). Named entity recognition experiments on Turkish texts. In *International Conference on Flexible Query Answering Systems* (pp. 524-535). Springer.

Lossio-Ventura, J. A., Jonquet, C., Roche, M., and Teisseire, M. (2014). Yet another ranking function for automatic multiword term extraction. *International Conference on Natural Language Processing*, 52–64.

Martin, W. J., Al, B. P., and Van Sterkenburg, P. J. (1983). On the processing of a text corpus: From textual data to lexicographical information. *Lexicography: Principles and practice*, 77-87.

Mason, O. J. (2006). *The Automatic Extraction of Linguistic Information from Text Corpora* [PhD Thesis, Birmingham University].

Maziarz, M., Szpakowicz, S., and Piasecki, M. (2015). A Procedural Definition of Multi-word Lexical Units. *Proceedings of the International Conference-Recent Advances in Natural Language Processing*, 427–435.

Mel'čuk, I. (1998). Collocations and lexical functions. *Phraseology: Theory, Analysis, and Applications*, 23–53.

Mel'čuk, I. (2006). Explanatory combinatorial dictionary. *Open Problems in Linguistics and Lexicography*, 225–355.

Moon, R. (1998). *Fixed Expressions and Idioms in English*.

Moon, R. (2008). Dictionaries and collocation. In S. Granger and F. Meunier (Eds.), *Phraseology: An interdisciplinary perspective*.

Nivre, J. (2021). Principles of the UD Annotation Framework. *Dagstuhl Seminar*, 98–99.

O'Donnell, M. B. (2011). The adjusted frequency list: A method to produce cluster-sensitive frequency lists. *ICAME Journal*, 35, 135–170.

Oflazer, K. (2014). Turkish and its challenges for language processing. *Language Resources and Evaluation*, 48(4), 639–653.

Passaro, L. C., and Lenci, A. (2016). Extracting terms with EXTra. *Computerised and Corpus-Based Approaches to Phraseology: Monolingual and Multilingual Perspectives*, 188–196.

Pawley, A., and Syder, F. H. (1983). Two puzzles for linguistic theory: Nativelike selection and nativelike fluency. In *Language and Communication* (pp. 203–239). Routledge.

Pearce, D. (2001). Synonymy in collocation extraction. *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 41–46.

Pecina, P. (2005). An extensive empirical study of collocation extraction methods. *Proceedings of the ACL Student Research Workshop*, 13–18.

Piao, S. S. L., Rayson, P., Archer, D., Wilson, A., and McEnery, T. (2003). Extracting multiword expressions with a semantic tagger. *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Volume 18, 49–56.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. *arXiv preprint arXiv:2003.07082*.

Ramisch, C., Villavicencio, A., Moura, L., and Idiart, M. (2008). Picking them up and figuring them out: Verb-particle constructions, noise and idiomaticity. *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, 49–56.

Ramisch, C., Villavicencio, A., and Boitet, C. (2010). Mwetoolkit: A framework for multiword expression identification. *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*, 662–669.

Ren, Z., Lü, Y., Cao, J., Liu, Q., and Huang, Y. (2009). Improving Statistical Machine Translation Using Domain Bilingual Multiword Expressions. *Proceedings of the 2009 Workshop on Multiword Expressions, ACL-IJCNLP 2009*, 47–54.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. *International Conference on Intelligent Text Processing and Computational Linguistics*, 1–15.

Schmitt, N., and Carter, R. (2004). Formulaic sequences in action. *Formulaic Sequences: Acquisition, Processing and Use*, 1–22.

Schone, P., and Jurafsky, D. (2001). Is knowledge-free induction of multiword unit dictionary headwords a solved problem? *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Shimohata, S., Sugio, T., and Nagata, J. (1997, July). Retrieving collocations by co-occurrences and word order constraints. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 476-481).

Shwartz, V., and Dagan, I. (2019). Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, *7*, 403-419.

Siepmann, D. (2005). Collocation, colligation and encoding dictionaries. Part I: Lexicological aspects. *International Journal of Lexicography*, 18(4), 409–443.

Sinclair, J. (2004). The search for units of meaning. In *Trust the Text : Language, Corpus and Discourse*.

Sinclair, J. M. (2008). The phrase, the whole phrase, and nothing but the phrase. In S. Granger and F. Meunier (Eds.), *Phraseology: An interdisciplinary perspective*.

Smadja, F. A. (1989). Lexical co-occurrence: The missing link. *Literary and Linguistic Computing*, 4(3), 163–168.

Smadja, F. (1993). Retrieving Collocations from Text: Xtract. *Computational Linguistics*, 19(1), 143.

Straka, M., and Straková, J. (2017). Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 88–99.

Stubbs, M. (2002). Two quantitative methods of studying phraseology in English. *International Journal of Corpus Linguistics*, 7(2), 215–244.

Stubbs, M. (2007). An example of frequent English phraseology: distributions, structures and functions. In *Corpus linguistics 25 years on* (pp. 87–105). Brill Rodopi.

Şeker, G. A., and Eryiğit, G. (2012). Initial explorations on using CRFs for Turkish named entity recognition. In *Proceedings of COLING 2012* (pp. 2459-2474).

Taşçıoğlu, T., and Kumova-Metin, S. (2021, June). Detection of Multiword Expressions with Word Vector Representations. In *2021 29th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

Trijp, Remi van. (2018, September 12). *Fillmore's Dangerous Idea*. http://www.essaysinlinguistics.com /2018/09/12/fillmore/

Tür, G., Hakkani-Tür, D., and Oflazer, K. (2003). A statistical information extraction system for Turkish. *Natural Language Engineering*, *9*(2), 181-210.

Uhrig, P., Evert, S., and Proisl, T. (2018). Collocation candidate extraction from dependency-annotated corpora: exploring differences across parsers and dependency annotation schemes. In *Lexical Collocation Analysis* (pp. 111–140). Springer.

Villavicencio, A., Bond, F., Korhonen, A., and McCarthy, D. (2005). Introduction to the special issue on multiword expressions: Having a crack at a hard nut. *Computer Speech and Language*, 19(4), 365–377.

Wahl, A., and Gries, S. T. (2020). Computational extraction of formulaic sequences from corpora. *Computational Phraseology*, 24, 83.

Wei, N., and Li, J. (2013). A new computing method for extracting contiguous phraseological sequences from academic text corpora. *International Journal of Corpus Linguistics*, 18(4), 506–535.

Wermter, J., and Hahn, U. (2004). Collocation extraction based on modifiability statistics. *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 980–986.

Wermter, J., and Hahn, U. (2006). You can't beat frequency (unless you use linguistic knowledge)--a qualitative evaluation of association measures for collocation and term extraction. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 785–792.

Wray, A., and Perkins, M. R. (2000). The functions of formulaic language: an integrated model. *Language and Communication*, 20(1), 1–28.

Wray, A. (2009). *Formulaic language and the lexicon*. Cambridge University Press.

Yeniterzi, R. (2011). Exploiting morphology in Turkish named entity recognition system. In *Proceedings of the ACL 2011 Student Session* (pp. 105-110).

Zhai, C. (1997). Exploiting context to identify lexical atoms--A statistical view of linguistic context. *arXiv preprint cmp-lg/9701001*.