Evaluation of automatic collocation extraction methods for language learning

Anonymous EMNLP submission

Abstract

A number of methods have been proposed to automatically extract collocations, i.e., conventionalized lexical combinations, from text corpora. However, the attempts to evaluate and compare them with a specific application in mind lag behind. This paper compares three end-to-end resources for collocation learning, all of which used the same corpus but different methods. Adopting a gold-standard evaluation method, the results show that the method of dependency parsing outperforms regex-over-pos in collocation identification. The lexical association measures (AMs) used for collocation ranking perform about the same overall but differently for individual collocation types. Further analysis has also revealed that there are considerable differences between other commonly used AMs.

1 Introduction

Collocations, as the most common manifestation of formulaic language, have attracted a great deal of research in the last decade (Wray, 2012). Most of the research on collocations has been connected to their definition (section 2.1) and extraction (section 2.2), but also to their acquisition, and consequently teaching. Herbst and Schmid (2014) argue, “Any reflection upon what is important in the learning and, consequently, also in the teaching of a foreign language will have to take into account the crucial role of conventionalized but unpredictable collocations. Any attempt by a learner to achieve some kind of near-nativeness will have to include facts of language such as the fact that it is *lay or set the table* in English, but *Tisch decken* in German, and *mettre la table* in French” (p. 1).

Collocation learning comes down to three main benefits for language learners: accurate production, efficient comprehension and increased fluency of processing (e.g., Men, 2017; Durrant and Mathews-Aydnih, 2011). Despite their widely recognized importance and ubiquity in language use, collocations pose a great challenge for language learners thanks to their arbitrary nature and the learner’s insufficient experience with the target language (Ellis, 2012). Thus, there is a pressing need to create resources for language learners to support their explicit collocation learning. Given the vast amount of collocations and the different goals of language learners, various methods have been proposed to extract them automatically from text. Yet it is still not conclusive which one performs the best for language learning and “the selection of one or another seems to be somewhat arbitrary” (González Fernández and Schmitt, 2015) (p. 96).

This paper attempts to evaluate three end-to-end resources of collocations built for language learning: Sketch Engine (Kilgarriff et al., 2014), Flexible Language Acquisition (FLAX) (Wu, 2010) and Elia (Bhalla et al., 2018, forthcoming). They use the same British Academic Written English (BAWE) corpus (Nesi, 2011), but different methods for collocation identification, i.e., regex-over-pos, n-grams combined with regex-over-pos and dependency parsing, respectively, and also different association measures for collocation ranking, i.e., Log Dice, raw frequency and Teaching Worth Formula (TWF). On top of that, we compare other widely used lexical association measures of MI, MI2, MI3, t-score, log-likelihood, Salience and Delta P using the data from the best performing candidate identification method as a baseline. For our evaluation, we use the expert-judged Academic Collocation List (ACL) (Ackermann and Chen, 2013) as a reference set (section 3.1), and calculate the recall and precision metrics.

1https://www.sketchengine.eu/
2http://flax.nzdl.org/greenstone3/flax
separately for collocation identification and ranking.

2 Theoretical Background

2.1 Notion of Collocation

Among the many different interpretations of collocations in the literature, three leading approaches can be distinguished: psychological, phraseological and distributional (Men, 2017).

The psychological approach envisages collocations as lexical associations in the mental lexicon of language users underlying their fluent and meaningful language use (e.g., Ellis et al., 2008). This perspective on collocations is supported by the evidence from psycholinguistic research using reaction time tasks, free associations tasks, self-paced reading and eye-tracking which suggests that collocations are holistically stored as chunks and thus processed faster (Wray, 2012). However, as found out by Meara (2009), the storage of word associations in the mental lexicon of native speakers is different from that of nonnative speakers.

The phraseological approach focuses predominantly on delimiting collocations (call a meeting) from free word combinations with a predictable meaning (call a doctor), on the one hand, and fixed idioms with an unpredictable meaning (call it a day) on the other (e.g., Cowie, 1998) by defining a set of criteria related to the compositionality of meaning and fixedness of form. Schmitt (2010) argues that such approach is rather problematic for the identification task as it is not clear how to operationalize such criteria without making it subjective and labor-intensive.

The distributional approach, also called Firthian or frequency-based, shifts the focus from the semantic aspects of collocations to structural. As Sinclair (1991) put it, “Collocation is the co-occurrence of two or more words within a short space of each other in a text. The usual measure of proximity is a maximum of four words intervening” (p. 170). Following this definition, various criteria have been considered for identifying collocation, e.g., distance, frequency, exclusivity, directionality, dispersion, type-token distribution and connectivity (Brezina et al., 2015). However, some researchers (e.g., Bartsch, 2004) argue that because of the little account of syntactic features of the words, it fails to capture certain collocations, e.g., the collocation collect stamps in the sentence They collect many things, but chiefly stamps, or vice versa, captures false collocations, such as things but.

Despite the obvious differences, there is considerable overlap between the three approaches as Durrant and Mathews-Aydin (2011) rightly point out, “Non-compositionality and high frequency of occurrence can both be cited as evidence for holistic mental storage, and non-substitutability of parts can be evidenced in terms of co-occurrence frequencies in a corpus” (p. 59).

2.2 Automatic Extraction of Collocations

The task of collocation extraction is usually split into two steps, that of candidate identification which automatically generates a list of potential collocations from a text according to some criteria, and that of candidate ranking, which ranks the list to keep the best collocations on top according to some association measure (Seretan, 2008).

2.2.1 Candidate Identification

In the candidate identification step, four prominent methods can be distinguished based on the proximity of words and the amount of linguistic information used: window, n-gram, regex-over-pos and parsing. The first two are based on linear proximity whereas the other two are on syntactic proximity.

The window-based method (e.g., Brezina et al., 2015) identifies collocations within a window of n words before and after the target word. It belongs to the most commonly known and used and directly follows the Firthian definition of collocations. Similarly, the n-gram method (e.g., Smadja and McKeown, 1990), extracts sequences of adjacent n words including the target word. The application of these two methods can vary along several dimensions, e.g., the nature of words considered, such as word forms, lemmas or word families (Seretan, 2008), the context span on the left and right or the number of grams, part-of-speech filtering, etc. However, due to the lack of linguistic information used, these methods are prone to many recall and precision errors (for a detailed discussion, see Lehmann and Schneider, 2009).

In contrast to the previous two methods, the regex-over-pos method takes into account the grammatical relations between words (e.g., Wu, 2010). It identifies collocations in text via regular expressions over part-of-speech tags which match a certain grammatical pattern of the collocation. An alternative, though less frequent, method iden-
tifies collocations in a syntactic relation via parsing (Seretan, 2008), and thus accounts for the syntactic flexibility feature of collocations. Bartsch and Evert (2014) found out that collocation extraction using parsing method improved the results in comparison to the window method. However, they also caution that the success depends on the accuracy of the parser and the set of grammatical relations used.

### 2.2.2 Candidate Ranking

The next step of candidate ranking entails measuring the strength of association between the two words, hence association measure (AMs). In principle, AMs compare the observed and expected frequencies of collocations in different ways, and thus differ in how much they highlight or downplay different features of collocations (for a detailed overview, see Pecina, 2010). There is no single best performing AM but rather the choice of an appropriate measure depends on the particular purpose and theoretical criteria. In language learning research and practice, the following AMs have received most attention: raw frequency, MI, MI2, MI3, Log Dice, t-score, log-likelihood, Salience, WTF and Delta P.

The Mutual Information (MI) measure prioritizes rare exclusivity of collocations which is strongly linked to predictability (Gablasova et al., 2017). However, it is also biased towards low-frequency combinations which can be circumvented by setting a minimum frequency threshold or giving extra weight to the collocation frequency by squaring (MI2) or cubing (MI3).

The Log Dice score is similar to MI2 and highlights the exclusivity of word combinations without putting too much weight to rare combinations. However, Log Dice, in contrast to MI2, is suitable for comparing scores from different corpora and has been described as a “lexicographer-friendly association score” (Rychlý, 2008, p. 6-9). Another measure adjusted for lexicographic purposes is Salience, the forerunner of Log Dice, which combines the strengths of MI and log frequency (Kilgarriff and Tugwell, 2002).

The t-score represents the strength of association between words by calculating the probability that a certain collocation will occur without considering the level of significance (Pecina, 2010). It prioritizes the frequency of the whole collocation, and hence there is a tendency for frequent collocations to rank higher.

The only measure created specifically for pedagogical purposes is the Teaching Worth Formula (TWF) which is again a combined measure of MI and the raw frequency with more weight given to the former. It was derived from an empirical research using both statistical measures and instructor judgments. Basically, the score represents “a prediction of how instructors would judge their teaching worth” (Simpson-Vlach and Ellis, 2010, p. 496).

In contrast to the previous measures, log-likelihood is a statistic which determines whether the word combination occurs more frequently than chance or not. In particular, the score does not provide information on “how large the difference is” but rather “whether we have enough evidence in the data to reject the null hypothesis” (Brezina et al., 2015, p. 161).

The last measure is Delta P which takes directionality into account and calculates the strength of the attraction between two words for each word separately. Therefore, in contrast to all the previous measures, it does not treat the collocational relationship as symmetrical (Gries, 2013).

### 3 Methodology

#### 3.1 Reference Set

The recently compiled Academic Collocation List (ACL) (Ackermann and Chen, 2013) was selected as the reference set (gold standard) to be compared against the test sets. Five main considerations drove this decision: First, it needed to be in line with the nature of the BAWE corpus that was chosen as a source input for extracting collocations. Since BAWE is a collection of academic writing of university students, the baseline set should also consist of academic collocations. Second, it should contain collocations consisting of two words as all the three resources focus on two-word collocations. Third, the collocations should preferably be grouped into collocation types based on their word classes or syntactic functions as in the test sets. Fourth, the reference set should be human-made or human-judged to ensure the quality of collocations. And finally, it should be compiled for pedagogical purposes.

The ACL comprises 2,469 lexical collocations in written academic English and is based on a written part of the Pearson International Corpus of Academic English (PICAE) of around 25 million words. It was carefully compiled using the
combination of automatic computational analysis and the numbers 1 and 2 indicate their positions in the collocation pair.

To make it comparable to the test sets, we lemmatized all its uninflected word forms using an automatic lemmatization tool from SpaCy \textsuperscript{3} and then manually checked all the errors. Next, it was organized by headwords with the POS tags \textit{noun}, \textit{adjective} and \textit{verb}, grouped by possible collocation types, and with the respective collocates appended resulting in a list of 1,455 headwords, 11 collocation types and with the numbers 1 and 2 indicate their positions in the collocation pair.

Table 1: Reference set grouped by collocation types starting with a headword where noun is \textit{n}, adjective is \textit{adj}, verb is \textit{v}, adverb is \textit{adv} and the numbers 1 and 2 indicate their positions in the collocation pair.

<table>
<thead>
<tr>
<th>Collocation Type</th>
<th>Headwords</th>
<th>Collocations</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1_n2</td>
<td>39</td>
<td>62</td>
</tr>
<tr>
<td>n2_n1</td>
<td>52</td>
<td>62</td>
</tr>
<tr>
<td>n2_v1</td>
<td>156</td>
<td>306</td>
</tr>
<tr>
<td>n2_adj1</td>
<td>483</td>
<td>1769</td>
</tr>
<tr>
<td>v1_n2</td>
<td>107</td>
<td>306</td>
</tr>
<tr>
<td>v1_adj2</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>v1_adv2</td>
<td>19</td>
<td>29</td>
</tr>
<tr>
<td>v2_adj1</td>
<td>79</td>
<td>139</td>
</tr>
<tr>
<td>adj1_n2</td>
<td>416</td>
<td>1769</td>
</tr>
<tr>
<td>adj2_v1</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>adj2_adv1</td>
<td>73</td>
<td>124</td>
</tr>
<tr>
<td>Total</td>
<td>1455</td>
<td>4626</td>
</tr>
</tbody>
</table>

Table 1: Reference set grouped by collocation types

3.2 Tests sets

3.2.1 Sketch Engine

Sketch Engine (SE) is an online corpus software with a wide range of functions and preloaded corpora which can be used for pedagogical purposes either indirectly, in the creation of textbooks and dictionaries, or directly in the classroom (Kilgariff \textit{et al.}, 2014). One of its functions is the Word Sketch for extracting collocations in a range of grammatical patterns, and one of its corpora is BAWE. The corpus is automatically POS-tagged using CLAWS \textsuperscript{4} 4 and the collocations are identified with the help of their embedded Sketch Grammar \textsuperscript{5} which is a set of regular expressions over POS tags. The retrieved collocates are then organized based on the grammatical relation to the headword and within each relation sorted by the Log Dice measure (alternatively, raw frequency).

The SE collocations were extracted using web scraping wherein, firstly, the URL was built using the lemma and POS tag of each word and then the eleven collocation types from the reference set were mapped to the collocation types used at SE to pickup the collocations of interest (Table 2). Picking up all the headwords (lemmas) from the reference set, the count and score of each collocate was stored in an intermediate file for each lemma in order to generate SE files \textsuperscript{6} for the final evaluation.

3.2.2 FLAX

FLAX (Flexible Language Acquisition) is an online library and tool specifically created for collocation learning (Wu, \textit{2010}). It consists of large collections of collocations and phrases extracted from different corpora, one of which is BAWE, and can be used for searching collocations for a particular word or for automatic generation of a variety of collocation exercises and games. The collocations are extracted using the combination of n-gram and regex over-pos methods which involved the following steps. Firstly, n-grams (n=5) are extracted from the corpus and tagged with the OpenNLP tagger \textsuperscript{7}. The tagged 5-grams are then matched against a set of regular expressions based on predefined collocation types \textsuperscript{8}. Finally, the individual collocations organized by collocation types are then sorted by raw frequency within each collocation type (Wu, 2010, p. 98).

For the evaluation, all FLAX collocations \textsuperscript{9}...

\textsuperscript{3}https://spacy.io/
\textsuperscript{4}https://www.sketchengine.eu/english-claws7-part-of-speech-tagset/
\textsuperscript{5}For a full list of Sketch Grammar, see https://www.sketchengine.eu/
\textsuperscript{6}Code and data in the ‘sketchengine’ folder of the Supplementary Material.
\textsuperscript{7}http://opennlp.apache.org/
\textsuperscript{8}For a full list of the collocation types with examples, see Wu \textit{et al.} (2010, p. 9).
\textsuperscript{9}Code and data in the ‘flax’ folder of the Supplementary Material.
Table 2: Mapping of collocation types between the reference set (ACL) and test sets (Sketch Engine, FLAX, Elia)

<table>
<thead>
<tr>
<th></th>
<th>Sketch Engine</th>
<th>FLAX</th>
<th>Elia</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1_n2</td>
<td>modifies</td>
<td>n-nn</td>
<td>NOUN + NOUN</td>
</tr>
<tr>
<td>n2_n1</td>
<td>modifier</td>
<td>n-nn</td>
<td>NOUN + NOUN</td>
</tr>
<tr>
<td>n2_v1</td>
<td>object_of</td>
<td>n-vn</td>
<td>VERB + NOUN</td>
</tr>
<tr>
<td>n2_adj1</td>
<td>modifier</td>
<td>n-an</td>
<td>ADJ + NOUN</td>
</tr>
<tr>
<td>v1_n2</td>
<td>object</td>
<td>v-vn</td>
<td>VERB + NOUN</td>
</tr>
<tr>
<td>v1_adj2</td>
<td>adj_comp</td>
<td>v-vppa</td>
<td>VERB + ADJ</td>
</tr>
<tr>
<td>v1_adv2</td>
<td>modifier</td>
<td>v-vr</td>
<td>VERB + ADV</td>
</tr>
<tr>
<td>v2_adv1</td>
<td>modifier</td>
<td>v-rv</td>
<td>VERB + ADV</td>
</tr>
<tr>
<td>adj1_n2</td>
<td>modifies</td>
<td>a-an</td>
<td>ADJ + NOUN</td>
</tr>
<tr>
<td>adj2_v1</td>
<td>adj_comp_of</td>
<td>a-vppa</td>
<td>VERB + ADJ</td>
</tr>
<tr>
<td>adj2_adv1</td>
<td>modifier</td>
<td>a-ra</td>
<td>ADV + ADJ</td>
</tr>
</tbody>
</table>

Table 2 clearly shows that the method of dependency parsing used by Elia resulted in higher overall recall (99%) than the method of regex-over-pos used by Sketch Engine (91%) and FLAX (84%). It seems that some dependency parsers have reached a sufficiently high accuracy to be used for collocation extraction or other NLP tasks (Levy et al., 2015). At the same time, there are obvious differences between Sketch Engine and FLAX, despite...
Looking closer at the results for Elia, we found out that exactly one half (19) of all the missed collocations (38) was due to parsing or tagging errors whereas the other half was due to different type classification; for example, the collocation learning activity was grouped under n2_adj1 in the reference set whereas, in Elia, it was assigned to n1_n2, and thus missed. This might as well be the case for some of the missed collocations in Sketch Engine and FLAX.

The precision, on the other hand, is very low for all (the highest 7% reached by Sketch Engine) at the expense of high recall. This, however, is not that important at this stage since the next step of ranking should shift all the irrelevant collocations to the bottom.

### 4.2 Candidate Ranking

For candidate ranking, recall and precision values were calculated for three samples of n-best candidates per headword for each test set: Top 4,626 where n refers to the exact number of collocates per each headword in the reference set, Top 14,550 to the 10-best collocates per headword, and Top 29,100 to the 20-best collocates per headword.

As illustrated in Table 4, the association measure Log Dice used by Sketch Engine performed slightly worse (36%) overall than Elia (40%) using TWF, a combination of MI and frequency, and FLAX (41%) using raw frequency for the Top 4,626 sample. As the sample increased to 14,550, Elia with a recall of 54% outperformed FLAX (52%) and Sketch Engine (51%). In the even larger sample of 29,100, Elia was still marginally better reaching 68% whereas Sketch Engine outperformed FLAX with a recall of 67% and 65% respectively. It seems that Log Dice improves its performance as more of the data is examined whereas raw frequency acts in quite the opposite way. However, it should also be pointed out that the differences between all of the scores are very subtle, less than 4% in all the samples. This is even more pronounced in the overall precision results which, for all three resources, are the same (17%) in Top 14,550 and almost the same (12%, 11%, 11%) in Top 29,100.

Looking at the individual CTs, an interesting picture of differences emerges. Sketch Engine’s measure performed consistently better for n1_n2, n2_n1, v1_n2 and v1_adj2 in all three samples. Elia’s measure performs consistently better for n2_v1 and adj1_n2. FLAX seems to perform better only for v2_adv1 and adj2_adv1 for Top 4,626 but it is not consistent for the other samples. Variability can be found not only among individual resources but also among individual CTs within one resource. For example, in Top 4,626, Sketch Engine reaches a recall of 19% for n1_n2 and of as high as 70% for adj2_v1. Recall values for Elia
range from 37% (n1,n2) to 97% (adj2,v1) and for FLAX from 7% (adj2,v1) to 86% (v1_adv2) in Top 14,550. The syntactic structure underlying collocations seems to have a great impact on the results, and thus should always be considered and specified as already suggested in some previous studies (e.g., Evert and Krenn, 2001; Bartsch and Evert, 2014).

To sum it up, despite the apparent similarities in the overall recall and precision values, it would be misleading to conclude that the three measures are equally efficient since they had a different data from the identification step to start with. It becomes clear when looking at the individual collocation types, for example adj2,v1 where Elia reached 43% as compared to 0% by FLAX. This could have been caused by the low recall (13%) of FLAX in the identification part. The issue with credit assignment is that it is not clear how much of the success can be attributed to the identification method discussed in the previous section and how much to the metric itself. To exclude the identification method as a factor, we decided to perform another analysis: the comparison of different AMs using the best-performing data from the candidate identification step, that is Elia, as a baseline to find out the differences when all things being equal.

4.2.1 Comparison of Different AMs with Elia as a Baseline

Using Elia collocations as a baseline, we have computed recall and precision for ten different ranking measures described in section 2.2.2. As in the previous section, we have computed it separately for the three samples of Top 4,626, Top 14,550 and Top 29,100, however, this time not for all individual CTs separately. The formulas used to compute each of the different AMs can be found in Brezina et al. (2015, p. 169-170).

The results on candidate ranking, arranged progressively by the best measures in Figure 1, show that recall curves for all AMs increase whereas precision curves decrease with the increased sample sizes as expected. In terms of coverage, the best performing measures are t-score and log likelihood across all samples with the recall values of 42%, 56%, 70% and 42%, 55%, 59% respectively. They are followed by Salience and TWF with the same values of 40% 54% 68%. All of these four measures exhibit a consistent behavior increasing by about 14% with the increased samples. On the other hand, the raw frequency measure, even though reaching similarly high scores for Top 4,262 (41%) and Top 29,100 (70%), increases only by 1% for Top 14,550. The next two measures MI3 and Log Dice lag slightly behind with the scores of 36%, 50%, 64% and 36%, 49%, 64% respectively, consistently increasing by about
Figure 1: Comparison of different association measures (AMs) using Elia as a baseline.

14%. The MI2 score performs significantly worse with a recall of 23%, 36%, 51%. The most collocations are missed by the measures Delta P and MI both reaching only a recall of 3%, 10%, 21%.

The precision values defining quality of the collocations point to very similar tendencies with t-score and log likelihood reaching the highest precision in all three samples with the scores of 42%, 18%, 11% and 42%, 17%, 11% respectively and with the TWF and Salience measures right behind, both with 40%, 17%, 11%. MI3 and Log Dice performs about the same with 37%, 16%, 10% and 36%, 15%, 10% respectively. Again, the MI2 score misses significantly more collocations than the previous measures reaching a precision of 23%, 11%, 8%. Surprisingly enough, MI and Delta P, both reached the lowest precision score of 3% for all samples. Thus, it can be concluded that the sample size does not affect the precision of the MI and Delta P association measures which, in the case of MI, is consistent with the previous findings by Evert and Krenn (2001).

These results question the dominant role of MI for collocation extraction (Gablasova et al., 2017), at least for language learning purposes. It also questions the assumption that Log Dice is fairly similar to MI or MI2 as our results suggests that it is actually more similar to MI3 (Gablasova et al., 2017). Furthermore, Delta P did not fulfill the expectations as expressed by Gries (2013). However, in defense of Delta P, it must be pointed out that the reference list did not indicate the direction of attraction for collocations, which is the underlying assumption of the Delta P measure, which might be the reason for the poor results. On the other hand, there are only subtle differences between some of the best-performing measures, such as log likelihood and t-score or TWF and Salience.

5 Conclusion

The aim of this study was to evaluate three collocation learning resources, namely Sketch Engine, FLAX and Elia, all of which used the same corpus but different methods for collocation identification and different lexical association measures for collocation ranking.

The findings indicate that using dependency parsing (Elia) for collocation identification led to much better results than using regular expressions over tagged corpus (Sketch Engine and FLAX). However, the success does not depend on the specific method entirely, but also on the quality of the set of syntactic structures. Using the same method with differently designed collocation types might lead to very different results, as was the case for Sketch Engine and FLAX.

The evaluation of collocation ranking has revealed that, overall, some of the association measures perform equally well, such as t-score, log-likelihood, TWF (used by Elia) and Salience. Raw frequency (used by FLAX) was also found to perform well but acting inconsistently across different sample sizes. The Log Dice measure (used by Sketch Engine) worked best for the majority of individual collocation types in comparison to raw frequency and TWF. On the other hand, the widely used MI and newly introduced Delta P were relatively poor in comparison to other AMs, but exhibited consistency in precision across varying sample sizes.

It has also become apparent that there are considerable differences between individual collocation types, and therefore should always be considered as a factor in collocation extraction. However, a future line of work is required to substantiate the consistency of these results on different reference lists and corpora.
References


