Polish Word Sketches

Adam Radziszewski  
Wrocław University of Technology  
adam.radziszewski@pwr.wroc.pl

Adam Kilgarriff  
Lexical Computing Ltd,  
University of Leeds  
am@lexmasterclass.com

Robert Lew  
Adam Mickiewicz University  
rlew@amu.edu.pl

Abstract

Word sketches are one-page automatic, corpus-based summaries of a word’s grammatical and collocational behaviour. They were first used in the production of the Macmillan English Dictionary (Rundell 2002). At that point, word sketches only existed for English. Today, the Sketch Engine is available, a corpus tool which takes as input a corpus of any language and corresponding grammar patterns and which generates word sketches for the words of that language. It also automatically generates a thesaurus and ‘sketch differences’, which specify similarities and differences between near-synonyms. A web corpus of Polish was morpho-syntactically tagged and loaded into the Sketch Engine. We describe the Polish Sketch Grammar and show how the resulting word sketches can be used in lexicography and for other linguistic purposes. The results show that word sketches could significantly facilitate lexicographic work for Polish, as they have for other languages.

Keywords: Polish, lexical profiling, Sketch Engine, word sketch, concordancer, web corpus, collocation, thesaurus, morpho-syntactic description

1. Introduction

Word sketches are one-page automatic, corpus-based summaries of a word’s grammatical and collocational behaviour. Their value for lexicographic work in English and other languages, as well as the background of the use of corpora in lexicography, have been described elsewhere (Kilgarriff and Rundell 2002, Kilgarriff et al. 2004).

First, we introduce corpus query systems and the basic idea of word sketches. Next, we concentrate on the application of word sketches to the Polish language in the Sketch Engine software.

2. Corpus Query Systems

A variety of corpus query systems (CQSs) have been used to examine corpus evidence since the rise of the first electronic corpora. Starting with the ground-breaking COBUILD project, lexicographers have been using KWIC (Key Word In Context) concordances as their primary tool for finding out how a word behaves. Later, with the growth of corpora, lexical statistics had to be applied to manage the abundant data and highlight the most salient collocations. Today, state-of-the-art CQSs allow the lexicographer great flexibility in searching for phrases, collocates, grammatical patterns, sorting concordances according to a wide range of criteria, selecting subcorpora for searching in, say, only spoken text, academic text, or only fiction. Available systems include WordSmith (Scott 2008), MonoConc (Barlow 2000), the Stuttgart Corpus Workbench (CWB, Christ and Schulze 1994) and Poliqarp (Przepiórkowski et al. 2004).

2.1. Sketch Engine

2.1.1. Description

The Sketch Engine is a corpus query system which gives access to the familiar CQS functions: concordances for several types of queries (simple, lemma, phrase, word form and CQL), with an integrated context control filter (Fig. 1).

Fig. 1: The Sketch Engine concordancer query window

The interface includes a variety of viewing and sorting options, as shown in Fig. 2.

Fig. 2: Concordancer viewing and sorting options

However, the features of the Sketch Engine which are of special interest in this paper are not part of standard concordancing programs. These features include word sketches, sketch differences and a thesaurus. They are all fully integrated with standard concordancing.
2.1.2. Word sketches

To identify a word’s grammatical and collocational behaviour, the Sketch Engine needs to know how to identify words connected by a grammatical relation. This can be achieved in one of two ways.

The first possibility is to parse the input corpus, so that the information about which word-instances stand in which grammatical relations with which other word-instances is embedded in the corpus. Currently, dependency-based syntactically annotated corpora are supported. We need to mark heads of phrases in phrase structure trees.

In the second approach, the input corpus is loaded into the Sketch Engine part-of-speech-tagged but not parsed, and the Sketch Engine supports the process of identifying grammatical relation instances through a sketch grammar. Grammatical relations (or grmrels, for short) will be defined one by one, using the Sketch Engine to test and develop them. Once the developer is satisfied with the definition of each grammatical relation, they save the file and the Sketch Engine then compiles it, finding all instances of all grammatical relations in the corpus. It puts them in a grmrels database, which gives users access to word sketches.

2.2. Grammatical relations

Grammatical relations are defined as regular expressions over part-of-speech (POS) tags, using the CQL formalism as first specified within the Stuttgart Corpus Wordbench and extended in Jakubíček et al (2010). For example, if we wish to include the grammatical relation between a noun subject and a passive participle after a copula, we capture a noun in the nominative ("subj:...nom:*"), any token of the być lemma (‘to be’), and a past participle in the nominative ([tag="ppas:...nom:*"]), also allowing for optional strings of intervening adverbs. Thus, we would end up with a definition along these lines (simplified for clarity):

```
=passive/subj_of_passive
1: [tag="subj:...nom:*"] [tag="adv:...*"] {0,3}
2: [lemma="być"] [tag="adv:...*"] {0,3}
```

The first line, starting with =, gives two names for the grammatical relation. The first name ("passive") applies when the arguments are in the order defined below, and the other grmrel name ("subj_of_passive") holds when the order of the arguments is reversed.

The numbers 1: and 2: mark the words to be extracted as the first and second arguments. [, , ], and * are standard regular expression metacharacters. \{0,3\}

---

1 The actual tags employed for Slavic languages usually include a lot more than just parts-of-speech, hence morpho-syntactic tagging or morpho-syntactic description (MSD) tagging would be a more suitable term. We use the traditional notion of POS-tagging here in a generic manner (the Sketch Engine treats the tags as strings anyway).

2 Full documentation is available at http://trac.sketchengine.co.uk/wiki/SkE/CorpusQuerying
allows for explicit attribute equality tests (e.g. 1:case=2:case). Such expressions can only be used if the corpus has been encoded with those attributes (case, number, gender) as additional features. The Sketch Engine contains a mechanism for runtime computation of these attributes from another, and we are in the process of exploring how this might be applied to Polish and the IPIC tagset. Without that mechanism, to handle such agreements involves enumerating every possible combination of number, gender and case values whenever an agreement is to be checked. As there are no less than 70 such combinations, we opted for a compromise, where only case value is checked (7 values).

Another challenge is the free word order typical of Slavic languages. One specific consequence of this is that the order of verb objects may vary, making it harder to capture the relation between a verb and its object, where the object is required to appear in a particular grammatical case. For instance, the expression “dal psu mioso” (“he gave the dog meat”), consisting of a verb followed by a dative object (“to the dog’) and one in the accusative (“meat”), be reordered (“dal mioso psu”) without any change in propositional meaning (if we disregard issues of topic/comment). We solved this problem by allowing for an optional intervening string of adjectives or nouns with the value of case other than the one involved. We also plan to explore the recent query language development with the within and containing operators (Jakubicek et al 2010).

The frequent references to parts of POS-tags in the relation definitions would make the grammar hard to read and maintain. Fortunately, the query language supports macro definitions (using the m4 POSIX standard). We make extensive use of the latter feature, providing some degree of abstraction over details related to the IPIC tagset. For instance, the macro N(case) is used to denote any noun or noun-like token bearing the given case.

2.4.1. Symmetric Example

We provide one symmetric relation, namely the coordination of two nouns, as well as two-word coordinate structures of the form “ani X, ani Y” (‘neither X nor Y’). Below is the top portion of the definition covering nominative and genitive cases (the remainder deals of the other grammatical cases in the same fashion):

```plaintext
=coord
*SYMMETRIC
1:N(nom) CONJ (N(nom) CONJ){0,5} 2:N(nom)
[word = "(?i)ani"] 1:N(nom) [word = ",",]?
[word = "ani"] 2:N(nom)
1:N(gen) CONJ (N(gen) CONJ){0,5} 2:N(gen)
[word = "(?i)ani"] 1:N(gen) [word = ",",]?
[word = "ani"] 2:N(gen)
```

The result of this grammatical relation can be viewed as part of the word sketch for the lemma nós (‘knife’) in Fig. 3.

The output shows that in the Polish corpus, 331 instances of this particular grammatical relation can be found for nós. By default, lemmas are ranked according to the salience score (Rychlý 2008), as illustrated in the left-hand pane above, but the user can change this to raw frequency. A number of other parameters can also be adjusted, either on a per-query basis or more permanently.

One possibility, illustrated by the alternative pane (right-hand column in our example) is allowing the Sketch engine to cluster the items, which may significantly facilitate lexicographic or linguistic description. In our example, the top coordinated item widelec (‘fork’) is highly predictable, the two items forming (just as they do in English, albeit in the reverse order), a conventionalized coordinate construction known as a freeze. The second most salient (and frequent) item siekiera (‘axe’), however, is hard to intuit. A quick examination of the remaining coordinate partners of nós reveals that they pattern into three groups: cutlery, hand weapons, and hand tools. The user can click on the number next to any lemma to see the relevant concordance, providing instances of use for lexicographic or lexical exploration.

![](image)

Fig. 3: Word sketch for the lemma nós (‘knife’)

2.4.2. Dual Example

Dual relations are most common. We defined 14 of them, including noun–modifier, subject–verb and verb–object relations. Following the grammars for Czech, Russian and Slovene, we defined multiple verb–object relations, each corresponding to one value of grammatical case of the object required by the verb. For instance, the
relation between a verb and its accusative object was defined as follows:
* DUAL
  = is_obj4/has_obj4
  2: VERB NV_SAFE_OTHER(acc)(0,5) 1:N(acc)
  1:N(acc) NV_SAFE_OTHER(acc)(0,5) 2: VERB

There are two variants of the relation, accounting for two orderings: the verb may be followed by the object (the unmarked order), or the object may come first (less frequent but possible). Note the use of a parameterized macro labelled NV_SAFE_OTHER. This expression is defined to capture forms that are likely to occur between a verb and its argument — e.g. adverbs, particles — as well as elements belonging to other arguments of the same verb (that is, nouns and adjectives bearing cases other than the one specified as the parameter). The measure allows us to provide compact definitions of a verb–object relation where word order is not fixed and a particular value of case is expected.

<table>
<thead>
<tr>
<th>prec verb</th>
<th>5768</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>skinać</td>
<td>325</td>
<td>10.64</td>
</tr>
<tr>
<td>pokwać</td>
<td>302</td>
<td>10.57</td>
</tr>
<tr>
<td>kwać</td>
<td>259</td>
<td>10.34</td>
</tr>
<tr>
<td>pokrćić</td>
<td>229</td>
<td>10.16</td>
</tr>
<tr>
<td>klinać</td>
<td>176</td>
<td>9.82</td>
</tr>
<tr>
<td>potrząsnąć</td>
<td>127</td>
<td>9.38</td>
</tr>
<tr>
<td>krćić</td>
<td>155</td>
<td>8.99</td>
</tr>
<tr>
<td>odwrócīć</td>
<td>166</td>
<td>8.99</td>
</tr>
<tr>
<td>zawracać</td>
<td>103</td>
<td>8.96</td>
</tr>
<tr>
<td>pochołyć</td>
<td>102</td>
<td>8.88</td>
</tr>
<tr>
<td>podnieść</td>
<td>276</td>
<td>8.86</td>
</tr>
<tr>
<td>spuśćić</td>
<td>87</td>
<td>8.63</td>
</tr>
<tr>
<td><strong>odwracać</strong></td>
<td>85</td>
<td>8.52</td>
</tr>
<tr>
<td><strong>podność</strong></td>
<td>123</td>
<td>8.18</td>
</tr>
<tr>
<td><strong>unieśće</strong></td>
<td>72</td>
<td>8.16</td>
</tr>
<tr>
<td><strong>potrząsać</strong></td>
<td>49</td>
<td>8.04</td>
</tr>
<tr>
<td><strong>chować</strong></td>
<td>63</td>
<td>8.02</td>
</tr>
<tr>
<td><strong>schylić</strong></td>
<td>41</td>
<td>7.79</td>
</tr>
<tr>
<td><strong>pochołyć</strong></td>
<td>40</td>
<td>7.7</td>
</tr>
<tr>
<td><strong>boleć</strong></td>
<td>66</td>
<td>7.6</td>
</tr>
<tr>
<td><strong>zwłeśće</strong></td>
<td>31</td>
<td>7.43</td>
</tr>
<tr>
<td><strong>zaprzęgać</strong></td>
<td>32</td>
<td>7.37</td>
</tr>
<tr>
<td><strong>chylić</strong></td>
<td>31</td>
<td>7.33</td>
</tr>
<tr>
<td><strong>wychylić</strong></td>
<td>30</td>
<td>7.29</td>
</tr>
<tr>
<td><strong>skłonić</strong></td>
<td>42</td>
<td>7.21</td>
</tr>
</tbody>
</table>

Fig. 4: Partial Word Sketch for the lemma głowa (‘head’)

Experience with other Slavic languages has shown, however, that often a simpler formula will generate very useful output. The example in Fig. 4 shows a partial Word Sketch for the lemma głowa (‘head’), where the relation prec_verb is simply defined as any verb following within a window of up to six items to the right. It will be seen that almost all the salient items identified exhibit useful patterns of głowa as object, and all represent structures which are lexicographically relevant due to their either central or additional metaphorical meaning. Thus, the concordance for the third item turns up many instances of “kwać głową” (‘nod one’s head’) in the literal sense, as well as examples of extended metaphorical use in the sense ‘shake one’s head (in disapproval)’.

2.4.3. Trinary Example

Trinary relations indicate the relations between three entities. In the Polish grammar they are used to extract patterns in which nouns and verbs combine with prepositional phrases. We provided two such relations: one to identify the material preceding the user-supplied form, the other to capture what follows it. Here is a fragment of the definition for the “preceding material” relation (again, the omitted part instantiates the same pattern with subsequent case values):
* TRINARY
  = prec_%s
  2:NOUN 3:[tag=’prep:nom.*’] NV_SAFE(0,5)
  1:N(nom)
  2: VERB 3:[tag=’prep:nom.*’] NV_SAFE(0,5)
  1:N(nom)
  2:NOUN 3:[tag=’prep:gen.*’] NV_SAFE(0,5)
  1:N(gen)
  2: VERB 3:[tag=’prep:gen.*’] NV_SAFE(0,5)
  1:N(gen)

In Fig. 5, the grammatical relation is illustrated between the lemma głowa preceded by the preposition na (‘on’), and this sketch snippet indicates salient combinations with nouns to the left. Along semantically transparent combinations with various type of headgear or hair (włos), we also capture the slightly specialized “PKB na głowę” ‘GDP per capita’.

<table>
<thead>
<tr>
<th>prec na</th>
<th>2063</th>
<th>4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>wlos</td>
<td>118</td>
<td>8.96</td>
</tr>
<tr>
<td>pkb</td>
<td>51</td>
<td>8.91</td>
</tr>
<tr>
<td>brutte</td>
<td>27</td>
<td>8.51</td>
</tr>
<tr>
<td>kaptur</td>
<td>44</td>
<td>8.38</td>
</tr>
<tr>
<td>chustka</td>
<td>25</td>
<td>8.33</td>
</tr>
<tr>
<td>chusta</td>
<td>19</td>
<td>8.3</td>
</tr>
<tr>
<td>korona</td>
<td>29</td>
<td>8.3</td>
</tr>
<tr>
<td>kapelusz</td>
<td>60</td>
<td>8.07</td>
</tr>
<tr>
<td>czapka 19</td>
<td>15</td>
<td>7.07</td>
</tr>
<tr>
<td>kask</td>
<td>10</td>
<td>7.06</td>
</tr>
<tr>
<td>chusteczka</td>
<td>10</td>
<td>7.06</td>
</tr>
</tbody>
</table>

Fig. 5: Sketch snippet for the lemma głowa (‘head’) followed by the preposition na (‘on’)

2.5. Sketch Differences

Synonyms (and antonyms) tend to share some of the collocutes but not all. The sketch differences module in the Sketch Engine highlights the shared and different collocational context of two specified words. The listing is colour-coded to show at a glance the commonalities and differences between the lemmas.
Our example in Fig. 6 shows a differential profile for two Polish near-synonyms meaning ‘hard, difficult’. The nouns in red (in the web interface; shaded and towards the top of each column, in the greyscale screenshot) exhibit collocational preference to trudny, whereas those in green (and towards the bottom of each column) to ciężki. Items against the white background collocate well with both adjectives. Sketch differences address the teasing apart of near-synonyms, one of the more difficult aspects of language description and use.

The reader may be surprised to see zadanie. This represents the nominal gerund “zadanie”. We have a recurring problem with participles and gerunds. which TAKIPI lemmatizes to their infinitival form. This, in conjunction with Takipi’s policy of leaving noun/gerund ambiguities unresolved, means that some gerunds end up lemmatized as infinitives (while some others exist independently as nominative nouns). This, in turn, leads to infinitive forms cropping up as lemmas in contexts where nouns are expected. The authors have had similar problems with gerunds (and also adjectival participles) for all the other languages where they have been involved in lexicography. Deciding when a particular gerund or adjectival participle should have its own dictionary entry is a recurring problem for human lexicographers, too.

2.6. Thesaurus

In this last feature of the Sketch Engine to be discussed here, word similarity is based on shared triples. In our example thesaurus entry for złośliwy (‘malicious’, Fig. 7), this adjective and ironiczny (‘ironic’) both occur as the third term in the triple <modifier, "usmiech", ?>, that is, both modify the noun ‘smile’). This provides one small piece of evidence that the two words are close in meaning. By pooling together all such pieces of evidence and weighting them according to salience (following the method developed by Lin (1998)), we identify the near neighbours for each word in a ‘distributional thesaurus’.

Comparing the output of the Sketch Engine with a leading dictionary of Polish synonyms (Dańrótka et al. 1993: 6), we find a degree of overlap, but the Sketch Engine also locates some synonyms missing from the dictionary (e.g. wrodný, gorzki). As is to be expected, the shared-triples approach also returns some antonyms (tagodny, ‘benevolent’) as well as words related in less simple ways.

Each of the items can be clicked on, which will take the user to a sketch difference for the two words, allowing a more in-depth exploration of the relations.

3. Conclusion and further work

We have presented a sketch grammar for Polish and shown how, through the Sketch Engine, it can be used to explore the grammar and lexis of Polish. The grammar is far from perfect and the project is ongoing, with short-term goals including a more elegant handling of case, number and gender agreement. As the output of the Sketch Engine is only as good as the underlying corpus, another agenda item is the preparation of a new and improved corpus. Despite these limitations, we believe that Polish word sketches are already a resource that is well able to support Polish lexicography and linguistic research in ways that have not been possible before.

### 4. Acknowledgements

Work on the Polish corpus was partially funded with the EACEA LLP KELLY grant, project number 505630-LLP-2009-1-SE-KA2-KA2MP.
5. References


