

Polish Word Sketches

Adam Radziszewski

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

Adam Kilgarriff

Lexical Computing Ltd,
University of Leeds
adam@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 (Kilgarrieff and Rundell 2002, Kilgarrieff 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).

Query Type: Lemma

Lemma: Simple
Lemma
Phrase
Word Form
CQL

Context

Lemma filter


Window: both 5 tokens.

Lemma(s): all of these items.

Make Concordance Clear All

Fig. 1: The Sketch Engine concordancer query window

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



Concordance Word List Word Sketch Thesaurus Sketch-Diff

Save View options Sort Sample Filter Frequency Collocations ConcDesc

Corpus: **Polish Web C**

Hits: 4

[Left](#) | [Right](#)

[Node](#)

[References](#)

[Shuffle](#)

<http://www.iberycopol.pl/1991/1991rat229.html>

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 *gramrels*, 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 *gramrels* database, which gives users access to word sketches.

2.2. Grammatical relations

Grammatical relations are defined as regular expressions over part-of-speech (POS) tags¹, using the CQL formalism as first specified within the Stuttgart Corpus Wordbench and extended in Jakubiček 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 ("subst:...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="subst:...nom:*"] [tag="adv:.*"] {0,3}
  [lemma="być"] [tag="adv:.*"] {0,3}
2:tag="ppas:...nom:*"]
```

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 *gramrel* 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).

indicates that the preceding term occurs between zero and three times.²

2.3. Polish Corpus

The Polish sketch grammar was developed and tested on a corpus of Polish. The corpus was gathered from the web by Serge Sharoff using the procedure described in Sharoff (2006). It contains 128 million tokens.

For the corpus to be loaded into the Sketch Engine, it must be POS-tagged and lemmatized beforehand. POS-tagging is the task of deciding the correct word class for each word in the corpus; e.g. determining whether the token "należycie" is an adverb ('properly'), or a form of the Polish verb *należać* ('to belong') in the 2nd person plural. A tagger presupposes a linguistic analysis of the language which has given rise to a set of the morpho-syntactic categories of the language (a tagset). Typically, a tagger also performs lemmatization, that is the assignment of lemmas. In our example, for the verb reading this would be the lemma *należać*, represented conventionally by the infinitival form.

The Polish web corpus was tagged and lemmatized by Lexical Computing Ltd using the TaKIPI tagger (Piasecki 2007). The tagger employs the tagset of the IPI PAN Corpus (IPIC), where 32 grammatical classes and their respective categories (e.g. number, gender and case for nouns) define the set of possible tags. Unlike in the traditional division into POS categories, the grammatical classes in the tagset are defined on the grounds of morpho-syntax with almost no reference to semantics (Przepiórkowski 2004). The tagset also defines the lemmatization decisions, that is which forms are considered lemmas for a particular approach to grammatical analysis.

The Polish sketch grammar is based on already existing grammars for Slavic languages: Czech, Russian and Slovene (Kilgariff et al 2004; Khokhlova 2010; Krek and Kilgariff 2006). Drawing on those definitions, we made some additional effort to account better for free word order.

The grammatical relations we defined for Polish include three types: **symmetric**, between two items with equal status; **dual**, between two dependent items; and **trinary**, involving three dependent items.

2.4. Regular Expressions and Slavic Morpho-syntax

The rich inflection found in Slavic languages is reflected in the complexity of the IPIC tagset. This, combined with free word order, makes it a challenge to capture syntactic relations in a word-order-based formalism..

A major technical challenge was to express the obligatory agreement in number, gender and case between Polish nouns and their adjective modifiers. The query language used in defining the grammatical relations

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 “dał psu mięso” (‘he gave the dog meat’), consisting of a verb followed by a dative object (‘(to) the dog’) and one in the accusative (‘meat’), may be reordered (“dał mięso 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 with the other grammatical cases in the same fashion):

```
=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 *nóż* (‘knife’) in Fig. 3.

The output shows that in the Polish corpus, 331 instances of this particular grammatical relation can be found for *nóż*. 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 *nóż* 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 lexicological exploration.

coord	331	3.1	coord	331	3.1
widelec	<u>40</u>	11.02	widelec	<u>40</u>	<u>47</u> 11.02
siekiera	<u>16</u>	9.35	tyżka <u>7</u>		
kastet	<u>7</u>	9.33	siekiera	<u>16</u>	9.35
nożyczki	<u>7</u>	8.94	kastet <u>7</u>	<u>11</u>	9.33
tasak	<u>4</u>	8.43	tasak <u>4</u>		
tyżka	<u>7</u>	8.3	nożyczki	<u>7</u>	8.94
pałka	<u>8</u>	8.14	pałka <u>8</u>	<u>14</u>	8.14
bagnet	<u>4</u>	7.76	kij <u>6</u>		
rewolwer	<u>4</u>	7.76	bagnet	<u>4</u>	7.76
młotek	<u>4</u>	7.68	rewolwer <u>4</u>	<u>12</u>	7.76
pistolet	<u>8</u>	7.56	pistolet <u>8</u>		
sztylet	<u>4</u>	7.42	młotek	<u>4</u>	7.68
hak	<u>4</u>	7.32	sztylet <u>4</u>	<u>8</u>	7.42
kij	<u>6</u>	7.08	miecz <u>4</u>		
igła	<u>4</u>	6.9	hak	<u>4</u>	7.32
miecz	<u>4</u>	5.31	igła	<u>4</u>	6.9
narzędzie	<u>5</u>	3.82	narzędzie <u>5</u>	<u>9</u>	3.82
broń	<u>4</u>	3.66	broń <u>4</u>		

Fig. 3: Word sketch for the lemma *nóż* (‘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.

prec verb	5768	2.0			
skinać	<u>325</u>	10.64	odwracać	<u>85</u>	8.52
pokiywać	<u>302</u>	10.57	podnosić	<u>123</u>	8.18
kiwać	<u>259</u>	10.34	uniesć	<u>72</u>	8.16
pokręcić	<u>229</u>	10.16	potrząsać	<u>49</u>	8.04
kiwnąć	<u>176</u>	9.82	chować	<u>63</u>	8.02
potrząsać	<u>127</u>	9.38	schylić	<u>41</u>	7.79
kręcić	<u>155</u>	8.99	pochylać	<u>40</u>	7.7
odwrócić	<u>166</u>	8.99	boleć	<u>66</u>	7.6
zawracać	<u>103</u>	8.96	zwiesić	<u>31</u>	7.43
pochylić	<u>102</u>	8.88	zaprzątać	<u>32</u>	7.37
podnieść	<u>276</u>	8.86	chylić	<u>31</u>	7.33
spuścić	<u>87</u>	8.63	wychylić	<u>30</u>	7.29
			skłonić	<u>42</u>	7.21

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 "kiwać 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'.

prec na	2063	4.0
włos	<u>118</u>	8.96
pkb <u>44</u>	<u>51</u>	8.91
brutto <u>7</u>		
kaptur	<u>27</u>	8.51
chustka <u>25</u>	<u>44</u>	8.38
chusta <u>19</u>		
korona	<u>39</u>	8.3
kapelusz <u>26</u>	<u>60</u>	8.07
czapka <u>19</u> hełm <u>15</u>		
kask	<u>10</u>	7.07
chusteczka	<u>10</u>	7.06

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 collocates 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.

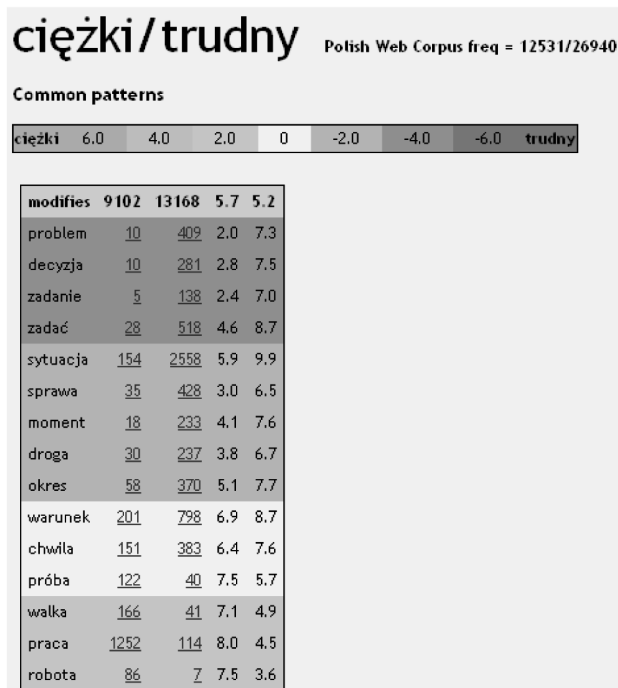


Fig. 6: Sketch difference: *ciężki* versus *trudny*

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 *zadać*. 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, “uśmiech”, ?>, 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 (Dąbrowka et al. 1993: 6), we find a degree of overlap, but the Sketch Engine also locates some synonyms missing from the dictionary (e.g. *wredny*, *gorzki*). As is to be expected, the shared-triples approach also returns some antonyms (*łagodny*, ‘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.

złośliwy

Polish Web Corpus freq = 1445

Lemma	Score	Freq	Cluster
<i>ironiczny</i>	0.215	490	<i>pogardliwy</i> [0.118, 171] <i>głupawy</i> [0.103, 113] <i>kpić</i> [0.1, 355] <i>drwić</i> [0.092, 324] <i>sztyderczy</i> [0.09, 161] <i>fielarny</i> [0.087, 68]
<i>przewrotny</i>	0.15	427	
<i>życzliwy</i>	0.143	1736	<i>szczerzy</i> [0.084, 3281]
<i>okrutny</i>	0.141	2416	<i>brutalny</i> [0.093, 1971]
<i>głupi</i>	0.137	5141	<i>sympatyczny</i> [0.091, 1784] <i>zabawny</i> [0.081, 1966]
<i>wredny</i>	0.134	446	
<i>złotański</i>	0.128	385	
<i>łagodny</i>	0.123	2424	
<i>niewinny</i>	0.118	2474	
<i>gorzki</i>	0.114	1247	
<i>cyniczny</i>	0.113	514	
<i>zjadliwy</i>	0.107	151	
<i>niewybredny</i>	0.105	110	<i>rubaszny</i> [0.089, 85] <i>rysunkowy</i> [0.081, 242]
<i>groźny</i>	0.105	3700	<i>niebezpieczny</i> [0.09, 7317]
<i>nowotworowy</i>	0.105	765	
<i>inteligentny</i>	0.103	2649	<i>bystry</i> [0.079, 661]
<i>agresywny</i>	0.1	2124	
<i>dziwny</i>	0.1	12800	<i>nieznany</i> [0.086, 4793] <i>taimniczy</i> [0.083, 3356] <i>pierwotny</i> [0.082, 6076] <i>zły</i> [0.081, 37438] <i>jakie</i> [0.08, 92002]

Fig. 7: Thesaurus entry for *złośliwy* (‘malicious’)

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

- Barlow, M. (2000). MonoConc Pro (Concordance software). Houston: Athelstan.
- Christ, O. and M. Schulze. (1994). The IMS Corpus Workbench: Corpus Query Processor (CQP) User's Manual University of Stuttgart. <http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/>
- Dąbrówka, A., E. Geller, R. Turczyn (1993). *Słownik synonimów*. Warszawa: MCR.
- Jakubiček, M., A. Kilgariff, D. McCarthy and P. Rychlý (2010). Syntactic searching in very large corpora for many languages. In: Otoguru, R. et al. (eds.). *Proceedings of Workshop on Advanced Corpus Solutions, PACLIC 24*.
- Khokhlova, M. (2010). Building Russian Sketches as Models of Phrases. In: Dykstra, A. and T. Schoonheim (eds.), *Proceedings of the XIV Euralex International Congress*. Ljouwert: Afûk. 364–371.
- Kilgariff, A. and M. Rundell (2002). Lexical profiling software and its lexicographic applications - a case study. In: Braasch, A. and C. Povlsen (eds.). *Proceedings of the Tenth EURALEX International Congress, EURALEX 2002*. Copenhagen: Center for Sprogteknologi, Copenhagen University. 807–818.
- Kilgariff, A., P. Rychlý, P. Smrž and D. Tugwell (2004). The Sketch Engine. In: Williams, G. and S. Vessier (eds.). *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004*. Lorient: Université De Bretagne Sud. 105–116.
- Krek, S. and A. Kilgariff (2006). Slovene Word Sketches. In: Erjavec, T. and J. Žganec Gros (eds.). *Proceedings of 5th Slovenian and 1st international Language Technologies Conference 2006*. Ljubljana: Jožef Stefan Institute.
- Lin, D. (1998). Automatic retrieval and clustering of similar words. *COLING-ACL*, Montreal. 768–774.
- Piasecki, M. (2007). Polish Tagger TaKIPI: Rule Based Construction and Optimisation. *Task Quarterly* 11/1–2. 151–167.
- Przepiórkowski, A. (2004). The IPI PAN Corpus: Preliminary version. Institute of Computer Science, Polish Academy of Sciences. Warsaw.
- Przepiórkowski, A., Z. Krynicki, Ł. Dąbrowski, M. Woliński, D. Janus and P. Bański. (2004). A search tool for corpora with positional tagsets and ambiguities. In: M. Lino, M. Xavier, F. Ferreira, R. Costa, R. Silva (eds.). *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*. Lisbon: ELRA. 1235–1238
- Rundell, M. (ed) (2002). *Macmillan English Dictionary for Advanced Learners*. Macmillan Education.
- Rychlý, P. (2008). A Lexicographer-Friendly Association Score. In Sojka P. and A. Horák (eds.). *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008*. Brno: Masaryk University. 6–9.
- Scott, M. (2008). WordSmith Tools version 5, Liverpool: Lexical Analysis Software.
- Sharoff, S. (2006). Creating general-purpose corpora using automated search engine queries. In: Baroni, M. and S. Bernardini. (eds). *WaCky! Working papers on the Web as Corpus*. Bologna: Gedit. 63-98.