

Manatee, Bonito and Word Sketches for Czech

Pavel Rychly and Pavel Smrz

Faculty of Informatics, Masaryk University

Botanicka 68a, 60200 Brno, Czech Republic

{pary, smrz@fi.muni.cz}

Abstract

This paper deals with a newly designed and developed system *Manatee* that can be employed to manage corpora, especially extremely large ones with billions of words, and enables the efficient evaluation of complex queries and the computation of advanced statistics. The main functions of the tool are presented here, together with the introduction of its web-based graphical user interface, *Bonito*. The sophisticated statistical processing is demonstrated in an example of computing of Word Sketches. Special attention is paid to the definition of the word sketches for Czech and problems connected to its free word order.

1. Introduction

Corpora, as a kind of empirical data, play a crucial role in current linguistics. While the size of the corpora has increased from several million to hundreds of millions (cf. Brown Corpus and American GigaCorpus), the management of such a vast amount of data is undeniably complicated. Our corpus management system *Manatee* is able to deal with extremely large corpora and is able to provide a platform for computing a wide range of lexical statistics.

An ideal general-purpose corpus management tool should embrace the complete life cycle of a corpus. For text data, it should enable:

- text preparation – conversion from various formats, encodings, etc.;
- metadata management – integration of the information about the source of data, authors, topics, genre, ...
- tokenization – language-dependent determination of the elementary unit accessed, usually a word;
- corpus annotation – potentially ambiguous, manual and automatic tagging on morphological, syntactic, semantic and pragmatic levels
- efficient corpus storage – the storage requirements of the indexes needed for querying should be minimized as should the time required for their creation;

- concordancing – retrieving language data matching the user’s query;
- computation of statistics – searching for typical patterns in data, frequency distribution of various features, co-occurrence statistics, etc.

Moreover, the ideal corpus management tool should implement all these tasks independent of:

- the language - especially text preparation, tokenization and corpus annotation;
- the platform (efficient storage and retrieval of corpus data as well as demanding computation present a challenging task for a platform independent implementation).

Currently, no single piece of software meets all of these requirements. However, the corpus manager Manatee implements seven of these criteria and provides an appropriate platform for integrating the language- and annotation-dependent tasks carried out by external tools.

2. Manatee and Bonito

The corpus management system Manatee is based on the text indexing library FINLIB [6] that provides storage and retrieval mechanisms for corpus data based on an efficient implementation of inverted indexes, suffix arrays, etc. The system processes either the output of an external tokenizer or that of the simple internal one, that does not allow for any language-dependent rules, e.g. for English *don't* or Russian *уз-за*. There are two kinds of attributes that can be provided – **positional** that can be defined for each token or structure – and **structural** that denotes a structure in the text (e.g. a sentence, paragraph, document, etc.).

The input format of Manatee is the so-called **vertical text** (or **vertical**) as specified by the Stuttgart Corpus Tools (<http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/>). Each word is on a new line, and for each word, there can be a number of fields specifying further information such as POS-tags, lemmas etc. The fields are separated by tabs. Corpora in other formats, such as the XCES standard, can be easily transformed to **vertical**.

The main aim of Manatee is large corpus management: in the most recent version, “large” exceeded 10^9 words and was only limited by disk size. This is in various languages and encodings, with no limits on the annotation, the number of attributes or meta-information. The architecture is modular and flexible. For example, it can be extended to implement better compression of special data and more statistics. It also queries and computes statistics very rapidly. The storage mechanism of Manatee is very efficient with all the necessary indexes and the compressed text requiring approximately 20 % more space than the original text.

Manatee implements a powerful query language. It enables searches given by restrictions on any attribute, on any meta information, or on any of their combinations. A given query can be further refined by means of positive or negative filters that are applied on the current result. A

result of any query can be stored as a new subcorpus and all further processing (e.g. statistics) can be done on that.

The tool also allows the computation of advanced statistical characteristics of given corpus data. The multilevel frequency distribution for KWIC can be grouped by attributes or meta information tags. Various statistical functions are computed for collocations (see Figure 1 for an example of collocation statistics).

The screenshot shows the Manatee corpus manager interface. The browser address bar contains the URL: `prfns=3&cbgrfns=l&cbgrfns=s&cbgrfns=c&csortfn=s`. The interface has a menu bar (File, Edit, View, Tab, Settings, Go, Bookmarks, Tools, Help) and a toolbar with navigation icons. Below the toolbar are tabs for Home, Query, Sketch, Thes, Sk-Diff, Conc, Freqs, and Colls. The main content area is titled "Collocation candidates" and displays a table with the following data:

	<u>Freq</u>	<u>T-score</u>	<u>MI</u>	<u>MI3</u>	<u>log likelihood</u>	<u>min. sensitivity</u>	<u>salience</u>
Night	76	8.710	10.123	22.618	920.280	0.013	43.838
Midsummer	75	8.659	13.308	25.766	1286.812	0.012	57.458
wildest	58	7.615	12.959	24.675	956.320	0.010	52.618
bad	113	10.557	7.186	20.826	904.196	0.008	33.970
dream	34	5.794	7.306	17.481	277.270	0.006	25.764
impossible	33	5.681	6.505	16.594	232.653	0.005	22.745
stuff	32	5.592	6.440	16.440	222.741	0.005	22.320
hopes	25	4.956	6.836	16.123	187.598	0.004	22.003
vivid	25	4.989	8.890	18.178	258.975	0.004	28.617
dreams	25	4.976	7.716	17.004	218.042	0.004	24.837
pipe	20	4.446	7.406	16.050	165.823	0.003	22.186
American	52	7.091	5.907	17.308	324.169	0.003	23.341
beyond	31	5.466	5.778	15.686	187.674	0.003	19.841
Sweet	18	4.237	9.510	17.850	202.137	0.003	27.488

Figure 1: Collocation candidates for the word *dream*

Other advanced features of Manatee that cannot be found in other corpus managers are multi-values, dynamic attributes and the definition of subcorpora. Corpus annotation is often ambiguous and the ambiguity cannot be easily/immediately removed. Manatee can process ambiguous annotation by means of multi-value attributes. For example, the POS tag for the Russian word *смать* will contain both noun and verb tags, and it will be displayed if one searches either for nouns or for verbs. Sometimes, it is easier to provide a program that generates an annotation on the fly than to include the full tagging in the corpus source (the annotation can be added, modified, extended, ...). Manatee defines these computed tags as dynamic attributes and is able to query and compute statistics as if they were static.

Manatee also functions as a corpus management server. Various interfaces have been defined to access its functions. There are command-line utilities to integrate the functionality in standard Unix-like pipelines, API (application program interface) for accessing of external tools, and graphical user interfaces (GUIs) – Bonito version 1 and 2. GUIs provide easy interfaces even for advanced functions such as dynamically generated forms for queries on any attribute, concordances. Bonito 1 is a standard multiplatform application that, once installed, can connect to the Manatee server and mediate most of its functions in a user-friendly way.

Bonito 2 is a new GUI that is completely web-based. Web pages are generated on the server (using CGI), with the standard web browser serving as the corpus client. It is easier for users – they do not need to familiarize themselves with a new application, and all the standard procedures such as cut & paste, bookmarks will work. There are also no security-policy problems with access through firewalls. Moreover, the web interface can be easily and promptly configured and extended as needed for various projects and/or user groups.

The known solutions of access and load control can be applied to the web server and the standard secured web protocol (https) can be used too. The localization of the tool can be simplified by page templates. Finally, it is easy to connect Bonito 2 to other sources and/or applications – the web pages can be read from other applications, and links to external sources (a picture, a sound sample, a video) can be presented.

3. Word Sketches for Czech

Manatee serves as a base for the Sketch Engine [4]. As it was defined in [3], Word Sketches is a short corpus-based summary of a word's grammatical and collocational behaviour (see Figure 2 for an example). The Sketch Engine takes as input a corpus of any language and corresponding grammatical patterns, and generates word sketches for the words of that language. It also generates a 'thesaurus' and 'sketch differences', which specify similarities and differences between semantically related words, e.g. near-synonyms like *clever* and *intelligent*.

unary rels	object_of 917 1.8	subject_of 395 1.5	a_modifier 1167 1.8	n_modifier 307 0.5
poss 1546 7.2	dream 40 39.51	come 83 31.39	wildest 57 58.47	pipe 22 29.2
Sfin 184 1.1	fulfil 33 32.28	haunt 8 20.77	bad 109 41.38	android 9 28.66
Swh 36 1.0	shatter 20 29.83	turn 14 15.09	vivid 23 31.19	anxiety 16 25.96
VPing 19 0.6	remember 29 21.59	fade 5 14.57	impossible 37 28.47	childhood 13 23.74
it+ 9 0.6	realise 20 21.27	end 9 14.16	sweet 23 26.57	Hollywood 9 21.51
VPto 58 0.5	haunt 10 20.94	seem 15 13.87	american 46 26.42	waking 6 21.34
Sing 7 0.4	recall 14 18.73	die 8 13.8	erotic 12 25.79	world 8 8.91
	destroy 13 17.07	become 16 13.15	weird 14 25.32	day 5 5.36
and/or 667 1.0	realize 11 16.97	begin 6 7.6	utopian 9 24.28	man 5 4.43
dream 40 32.31	report 18 15.96	go 9 7.47	recurring 10 24.17	
hope 33 28.66	analyze 5 15.39	take 7 3.98	wet 17 22.43	
fantasy 19 27.33	interpret 9 14.85		tangerine 5 20.16	
nightmare 13 23.56	abandon 9 14.52		terrible 14 19.81	
memory 21 21.43	harbour 5 14.32		strange 15 18.7	
ambition 13 20.82	dash 5 14.05		manifest 6 17.59	
aspiration 10 20.13	make 53 13.26		romantic 9 17.0	
vision 11 17.02	achieve 12 12.81		foolish 7 16.33	
illusion 6 15.43	share 10 12.59		lifelong 5 14.99	

Figure 2: Sketch for the word *dream*

It is relatively easy to define grammatical patterns for languages with a regular word order, such as English. On the other hand, languages with a relatively free word order can pose obstacles for non-conflicting sketch patterns. For example, the sentence structure of Slavic languages is designated as free, but the term serves only as a label for word order that cannot be easily described by a set of rules based only on syntactic criteria. The word order plays an important role in communicative dynamism in expressing the sentence focus. This phenomenon has been intensively examined by Prague Linguistic School in the context of Functional Generative Description [1], among others.

As discussed in [4], we have derived grammatical patterns for Czech in a stepwise process looking for a tradeoff between the precision and recall of the patterns. This procedure has demonstrated that most errors are due to the incorrect tagging of the corpus used.

We will demonstrate the definition of grammatical patterns for Czech through the following examples. Patterns are given in the form of queries in the standard Manatee format (usually produced by a macro-processor such as m4 from a more readable form). Patterns contain two labeled positions (1: and 2:). The 1: and 2: mark the words to be extracted as the first and

second arguments of the grammatical relation. Lines beginning with the equal sign (=) name the relation defined. Thus, the lines:

```
=post_inf
    1:[] 2:verb_inf
```

define `post_inf` relation for any word followed by infinitive (such as Russian *желание умереть*).

Lines beginning with an asterisk (*) are processing directives. They modify how the following lines are handled. *SYMMETRIC evaluates queries also with the `1' and `2' labels swapped. The pattern:

```
=coord
*SYMMETRIC
    1:[] [word = "a" | word = "nebo"] 2:[] & 1.k=2.k & 1.c=2.c
```

will extract all pairs with coordinate conjunctions *a* (and) or *nebo* (or) that have the same word class (`1.k = 2.k`) and an agreement in grammatical case (`1.c = 2.c`).

The keyword DUAL specifies that there are two relations defined on one line, e.g. *is_subj_of* and its converse, *has_subj*, and a single instance of the relation contributes an *is_subj_of* relation to the noun and a *has_subj* relation to the verb in the following example:

```
*DUAL
=is_subj_of/has_subj
    1:noun_nominative gap([NVZJP].*) 2:[verb_p3X & !aux_verb]
    1:noun_nominative gap([NVZJP].*) 2:[verb_passive & !aux_verb]
    2:[verb_p3X & !aux_verb] gap([NVZJP].*) 1:noun_nominative
    2:[verb_passive & !aux_verb] gap([NVZJP].*) 1:noun_nominative
```

The problem of free word order is addressed by the simple mechanism of gaps in these patterns. The object `gap()` matches up to 5 words differing in their categories from the given list.

*TRINARY is used for trinary relations. These are translated into regular binary relations with different names. A name of a trinary relation contains `%s' and respective queries contain the third label 3: A value of the attribute on the position labeled 3: is then substituted for `%s' in the relation name. Therefore, the following pattern:

```
*TRINARY
=post_%s
    1:verb 3:prep adj_string 2:noun & 3.c = 2.c
```

will produce several lists of nouns for each verb named according to the preceding preposition, such as `post_v`, `post_na`, `post_k` etc. in Czech.

As the current situation for Czech and Russian demonstrates, it is often easier to develop a robust morphological analyzer (cf. Dialing [5] and AJKA [7]) than to implement a POS tagger (disambiguator) for Slavic languages. In the process, we have explored the possibility of

computing word sketches from a morphologically tagged, but not disambiguated, Czech corpus.

Although we can not provide conclusive final results, our preliminary findings are very promising. The same texts that form the Czech corpus used in our previous experiments [4] have been morphologically tagged by AJKA and a new corpus has been loaded into Manatee. The grammatical patterns have been redefined to cover AJKA's tag set as well as the ambiguity of the annotation.

corpus	automatically disambiguated	ambiguously tagged
tags/word	1	4.7
time of processing	51 min.	253 min.
size of output	58.2 mil. triples	97.6 mil. triples
different triples	18.7 mil.	32.6 mil.
different lemmas	776 252	952 278
most salient triples	1.4 mil.	2.4 mil.
intersection in the most salient patterns	58 %	

Table 1: Quantitative characteristics of word sketches for Czech

The quantitative evaluation is shown in Table 1. The sketches resulting from the ambiguously tagged corpus are larger and one could worry how much noise is added to the output. Fortunately, there is a rather large intersection for the most significant patterns. The qualitative evaluation confirms these results. The misclassified words can be easily removed from the generated lists and the statistical nature of the sketches on the ambiguous input even eliminates peculiar errors that are present in the automatically disambiguated corpus.

4. Conclusions and Future Directions

The Sketch Engine as a commercial product has been launched only recently (Euralex 2004). Nonetheless, it is already being used in the construction of the Irish dictionary *Foras na Gaeilge* (sponsored by the Irish Government) and at Oxford University Press, where 'sketch difference' function is used for the new edition of Oxford Thesaurus of English. Manatee is employed as the main corpus management tool for several large corpora, including the Czech National Corpus.

The future directions of our work are as follows. Firstly, further technical upgrades of the implemented tools will aim at better portability: (1) full-featured Manatee-Bonito2 on MS Windows, support for 64-bit architectures, run from CD/DVD without requiring installation, (2) extensions of API: multi-parallel corpora, word-sense disambiguation,

syntactical dependency structures ..., and (3) a further simplification of the user interface: attribute names, explanation of tags, easy creation of a corpus, configuration and personalization. A new query language will be defined and implemented as well as advanced options of multi-level text tokenization, which involves filtering and joining tokens, queries and statistics on a given tokenization level. The web-based interface of Bonito2 also opens doors for the “web as a corpus” ideas [2].

Secondly, we are also currently working on the application of the relation patterns for automatic ontology acquisition. The interest in ontologies increased with the recognition of their importance for the Semantic Web. The emerging information systems need the definition of a common understanding for their application domains that should be given by ontologies. The precision of the semantic relations is often rather low, partly because the current methods often underestimate the need for linguistically processing the corpus as well as because rather limited corpora are often used. As our preliminary results show, a slightly modified version of the word-sketch patterns could help to overcome both these obstacles.

Acknowledgement

This work was supported by Ministry of Education of the Czech Republic Research Plan CEZ:J07/98:143300003 and by the Academy of Sciences of the Czech Republic “Information Society” program, the research grant 1ET100300419.

References

1. Hajicova, E., Sgall, P. and Skoumalova, H. An Automatic Procedure for Topic-Focus Identification. *Computational Linguistics* 21, 1994. pp. 81 – 94.
2. Kilgarriff, A. and Grefenstette, G. Web as Corpus: Introduction to the Special Issue. *Computational Linguistics* 29 (3). 2003.
3. Kilgarriff, A. and Rundell, M. Lexical Profiling Software and its lexicographic applications: a case study. Proc. EURALEX 2002, Copenhagen. 2002: 807-818.
4. Kilgarriff, A., Rychly, P. Smrz, P. and Tugwell, D. The Sketch Engine. Proc. EURALEX 2004, Lorient. 2004.
5. Leontieva, N. N. Automatic Analysis of Russian Morphology. (<http://www.aot.ru/>).
6. Rychly, P. Corpus Managers and their effective implementation. PhD Thesis, Faculty of Informatics, Masaryk University, 2000.
7. Sedlacek, R. and Smrz, P. A New Czech Morphological Analyser AJKA. Proc. TSD, Brno, Czech Republic, 2001, pp. 100 – 107.