Appears in: Prague Bulletin of Mathematical Linguistics, vol. 75, Univerzita Karlova, Praha, 2001

Parsing with Regular Expressions: A Minute to Learn, a Lifetime to Master

Daniel Zeman

Centrum komputační lingvistiky
Univerzita Karlova
Malostranské náměstí 25, 11800 Praha 1, Czechia
zeman@ufal.ms.mff.cuni.cz

Abstract

This paper reports on a work in progress, which tries to combine statistical modeling and manually written regular rules for parsing Czech. While the syntactic structure of a sentence may be very complex and will never be fully covered by a finite-state machine, there are phenomena such as noun phrases and simple coordinations that do quite well. Even these "chunks" may theoretically be unlimitedly complex but I show that in real world they rarely do. So a shallow finite-state parser can be built to preprocess the input text and solve the simple chunks without introducing too many errors. With a minimum of programming effort, such preprocessor currently covers almost 20% of input words with an accuracy of more than 90%. Further increasing of the coverage without loss of accuracy would require much more effort, so the rest is left for the statistical parser.

1. Introduction

This work was once motivated by the following question: "Why not do simple things a simple way?" I was working on a statistical parser, which itself is a relatively simple tool in comparison with, say, a grammar. The parser (see Zeman 1998, Hajič et al., 1998) is intended to parse Czech sentences and produce structures compatible with the analytical level (AL) of the Prague Dependency Treebank (PDT, see Böhmová et al.). More specifically, for each input word it has to find another word from the same sentence, that (according to the definition of AL) is its governor[1]. The parser is not lexicalized; it only relies on the morphological properties of words, encoded in morphological (POS) tags.

The main idea of the method the parser uses is as follows: it counts dependencies between words of particular morphological category (e.g. between `NNMS1---A------` (masculine singular noun in nominative case) and `AAMS1---A1-----` (masculine singular adjective in nominative case)). The table of relative frequencies of dependencies is then used to build the most probable dependency tree for a particular sentence. Although several supporting statistics are collected to improve the accuracy, the model remains one-dimensional in the sense that it only judges one relation (between two words — the governor and the dependent). There are however phenomena that are multidimensional in that sense. For instance, coordinations are modeled as pseudo-dependencies on the analytical level (see Figure 1). The coordinating conjunction is regarded the governor (head) of the coordination and the coordinated members are its dependents[2].

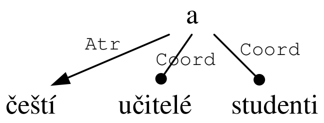

Figure 1 Dependency representation of the phrase *čeští učitelé a studenti* ("Czech teachers and students"). The edge between *čeští* and *a* is real dependency of the adjective on the coordination of nouns. The other two edges just build the coordination.

As a result, there are (pseudo) dependencies, which have to be compatible with other edges in the tree. The parser should not only know that a noun (an adjective, a verb…) may depend on a coordinating conjunction, it also should know that this noun usually agrees in case with the other coordinated nouns. Thus the dependency probability is here a function of at least three variables: the governor, the dependent, and the dependent's sister, if any.

Since the parser does not consider the sister nodes it usually fails to capture coordinations correctly. It learns that almost any kind of word can depend on a conjunction, and in child innocence it gathers nouns, verbs and all other stuff

[1] For technical reasons the AL definition requires that each word (token) have its own node in the dependency tree.

[2] Special tags — "analytical functions" — distinguish between coordination members and attributes of the whole coordination.

under one “and”. This problem can be solved in a few ways. Either the model can be extended to look at sisters. This is indeed possible although it may enlarge problems with data sparseness. But would the added complexity be adequate to the whole task? NP coordinations (as opposed to coordinations of verbs/clauses) are not too complex, they are just incompatible with the current model. What if they were processed in advance, replaced by a representative noun, and thus hidden from the statistical model? The parser would than deal with verb coordinations only, and hopefully it would be more successful.

I used regular expressions to model the local syntax of coordinations, simple noun phrases and other simple small chunks of sentences. The method is closely related to *local grammars* that have been applied to a similar task for a similar language (Serbo-Croatian; see Nenadić and Vitas 1998, Nenadić 2000).

2. Two ways of combining a shallow parser with a full one

This paper focuses on the shallow preprocessor; the results of combining it with a full parser will be published in future. However, before we build a preprocessor, we should think of the way its output will be used.

I mentioned one option in the introduction. The preprocessor would replace each recognized phrase by a representative word. For a normal phrase the head would be its representative. A coordination should be represented by one of the coordinated members, eventually converted to plural. The parser then does not see the preprocessed phrases neither during the training phase, nor during testing. Such phrases become atomic items for the parser.

There is a possible drawback of this method. If the preprocessor fails to find all members of a phrase, the error can be corrected by the parser only when the forgotten member depends on the head of the phrase. If it shall be nested more deeply in the phrase, its real governor is now invisible because the phrase is atomic. This leads to a second approach where the phrase would even for the parser be a structure rather than a monolith. The parser would get some dependencies for free but would be allowed to add new dependencies at any place in the structure. Note that this second approach is in contradiction with the original motivation — hiding coordinations from the parser. But once the preprocessor is built we should explore both ways to see which helps the parser better.

3. The preprocessor implementation

The preprocessor is written in Perl (see Wall et al., 1996). There is a procedure that scans the Prague Dependency Treebank and repeatedly applies a set of regular expressions (RE’s) to each sentence. The RE’s can be easily modified and new RE’s can be added. Every RE describes a local configuration that signalizes a dependency between two chunks of the input. A chunk is a word or a sequence of chunks that has been recognized by a previous application of an RE (and replaced by a representative word).

For instance, a coordination of adjectives is defined by the following three sub expressions:

```
"<t>A.([^X])([^X])(\\d)"
"<l>(a|i|nebo|ani)<"
"<t>A.\\1\\2\\3"
```

The Perl procedure combines these sub expressions into one RE. The resulting RE finds a sequence of words such that the description of the first word contains a match of the first sub-RE, the second word description matches the second sub-RE and so on. So in this case we are looking for a sequence of three words where the first one has a morphological tag beginning with A (adjective), having any character at second position (sub-part-of-speech), having known gender and number (unknown is encoded as X, which is prohibited by the RE), and having known case encoded as a digit. The second word must have the lemma *a* (“and”), *i* (“as well as”), *nebo* (“or”) or *ani* (“nor”). The third word must again be an adjective (the tag beginning with A) and must agree with the first one in gender, number and case. The agreement is enforced by the sub-RE’s \1, \2, and \3. They refer to the first, the second and the third parenthesized sub-RE’s, respectively. In our case all referred parentheses occurred in the first word’s tag, marking its gender, number and case.

The example RE would thus match for instance the coordination *přímým i nepřímým* (“direct as well as indirect”; instrumental case) that has the following (simplified) representation in PDT:

```
<f>přímým<l>přímý<t>AAMS7---A1-----
<f>i<l>i<t>J^-------------
<f>nepřímým<l>přímý<t>AAMS7---N1-----
```

On the other hand, the RE would not match the sequence *žlutého a černou* (“yellow / Masc. and black / Fem.”) because the adjectives do not agree in gender:

```
<f>žlutého<l>žlutý<t>AAMS4---A1-----
<f>a<l>a<t>J^-------------
<f>černou<l>černý<t>AAFS4---A1-----
```

Two special parameters are passed to the procedure. They are interpreted as relative indices of the phrase head and of the phrase representative (1 and 2 in our case, the first word has the index of zero).

Each RE is applied repeatedly to each sentence until all matches are found. This can also accommodate some recursive constructions. Suppose we have an RE for adjective-noun NP's. And suppose there is a noun preceded by two adjectives that both modify it, e.g. *velký zelený strom* ("big green tree"). First the sequence *zelený strom* is recognized and replaced by the representative noun *strom*. This way the words *velký* and *strom* become neighbors and the sequence *zelený strom* can be recognized in the next run.

To keep the process simple, the order in which different RE's are applied is fixed. Nevertheless some RE's can be tried at several points to see whether the replacements created new material for them. To illustrate this, let's have two RE's called AN (adjective-noun NP), and NaN (coordination of nouns), and let's apply them in the order NaN-AN-NaN. Then if the input contains the sequence *čeští sportovci a slovenští sportovci a umělci* ("Czech sportsmen and Slovak sportsmen and artists"), the first NaN reduces it to *čeští sportovci a slovenští umělci*, this sequence is further reduced by the AN to *sportovci a umělci*, which is finally recognized by the second NaN. This example shows that the covered constructions can be richer than one might think. On the other hand it also reveals the main weakness of the approach: in other configurations such as *čeští hráči i Němci* ("Czech players as well as the Germans"), the RE sequence AN-NaN would be appropriate. So the method can be relatively successful only if there exists an ordering of RE's that matches the data in much more cases than any other ordering.

The programming simplicity and efficiency is a significant issue. The driving procedure was finished in a few days, a regular expression can be formulated and added in a minute or two and the Perl interpreter applies it to a corpus of 19126 sentences often in less then one minute[3]. However, I am aware that solving the last 10 % of errors would consume 90 % of total effort and make the RE's too complex and unintelligible. That's why my ambitions are only to help the parser, not to replace it. And that also is why I borrowed the promotion slogan from the manufacturer of a board game called Othello: it takes a minute to learn but a lifetime to master.

4. Kinds of chunks covered

In this section I will introduce the particular regular expressions I tested. All the RE's were run on the training section of PDT 0.5 (19126 sentences). Note that using "training" data for evaluation is no crucial problem in this case because the only dependency of the RE's on the data is that I review the errors and eventually modify the RE's to avoid the errors.

4.1. Coordinations

To get a list of words that can serve as roots of coordinations[4] I ran through the training section of the corpus where dependencies and their labels are present. The following table shows for each token the number how many times it ruled a coordination (I list only the tokens that appeared more than once), and the number how many times it appeared with other function.

a	and	6557	282
,	,	2205	16763
ale	but	861	35
–	–	642	4495
nebo	or	518	10
:	:	442	1225
i	as well as	411	1109
však	however	384	282
či	or	253	1
proto	that's why	139	62
až-1	through	136	0
tak	so	95	439
/	/	80	161
neboť	because	73	1
ani	nor	51	213
až-2	up to	35	0
tedy	thus	35	230

[3] On a 266 MHz PC running Linux.

[4] The coordinating conjunctions can be recognized after their morphological tag but there are also other POS's that can serve as coordinating elements, e.g. a comma.

avšak	but	32	0
jenže	but	27	1
zatímco	while	27	19
přesto	though	25	18
nýbrž	but	23	0
ovšem	though	17	108
+	+	14	6
&	&	12	2
případně	as the case may be	11	17
and-1	and	11	0
ba	even	8	1
tudíž	consequently	8	3
respektive	respectively	8	8
(	(	8	1967
čili-1	or	7	0
jednak	as well as	7	12
.	.	7	18570
leč-2	however	6	0
plus	plus	6	3
nicméně	nevertheless	6	17
takže	so that	6	64
jenomže	but	5	0
kdežto	whereas	5	0
vždyť	yet	5	6
*	*	5	230
zato	yet	4	4
aneb	or	4	6
x	x	4	21
dokonce	even	4	94
;	;	4	143
et	et	3	0
přece	though	3	55
což-1	which	2	0
und	und	2	0
popřípadě	as the case may be	2	2
tj	i.e.	2	28
nikoliv	not	2	38
jinak	otherwise	2	58
	TOTAL	**13274**	

Not all listed coordinators occur in the same pattern. Some occur in pairs, many require that a comma precede them. I selected the four that occur in the simplest pattern, `"noun coord noun"`: *a, i, nebo, ani*. The pattern is described by two RE's, called NaN and AaA. The first RE models coordinations of nouns, the second coordinations of adjectives. AaA requires that the coordinated members agree in gender, number and case. NaN requires only agreement in case. Only simple coordinations of two elements were tested but an RE for larger coordinations (e.g. *Peter, Paul and Mary*) could easily be created and added. One RE could even capture coordinations of arbitrary (unlimited) size. However, large noun coordinations are quite rare in the real data, so even our toy RE can solve a lot.

Of course the coordination processor will be more efficient if we are able to process simple noun or adjective phrases first. See below the description of the respective RE's. I searched the text for DA's (adverb-adjective pairs) before AaA, and for AN's (adjective-noun pairs) before NaN.

The procedure found 522 good AaA's and as little as 4 bad (where *bad* means that a false dependency was proposed). This gives the accuracy of 99.2 %.

For NaN's, 2151 good and 114 bad ones were found; the accuracy is 95.0 %. The most frequent error occurred in coordinations of names (*George* ***Bush and Al*** *Gore* — the incorrectly recognized part of the phrase is printed in bold) or similar constructions (*předseda SNR Ivan* ***Gašparovič a velvyslanec*** *Izraele v ČSFR Yoel Sher* ("SNR chair Ivan Gašparovič and the Israeli Ambassador to CSFR, Yoel Sher")). This suggests a new RE that would combine sequences of nouns agreeing in gender, number and case.

4.2. Adjective – noun NP's

The following RE, called AN, recognizes noun phrases that consist of an adjective followed by a noun; the adjective agrees with the noun in gender, number and case. The noun is the head and the representative of the phrase.

```
"<t>A.([^X])([^X])(\\d)"
"<t>N.\\1\\2\\3"
```

I applied this RE to the text after AaA but no NaN had been applied. (Further tests shall be done to see whether earlier application of NaN removes more errors than it introduces.) Of the 28755 AN phrases found, 28410 were good and 345 bad, which gives the impressive accuracy of 98.8 %. Among the errors made, the following occurred periodically and deserve more careful handling:

1. Other kinds of coordinations, uncovered by my toy RE: *u První americko –* ***české pojišťovny*** ("at the First American – Czech Insurance Company"). The same holds for appositions.
2. NaN should have been applied first: ***českým zemědělcům*** *a potravinářům* ("Czech farmers and workers in the food industry").
3. Ordinal numerals behaving like adjectives should be handled as adjectives: *první a druhá vlna* ("the first and the second wave").
4. Nominal predicates in sentences with the OVS word order (the adjective *ideální* shall depend on the auxiliary verb *jsou*): *pro ně jsou* ***ideální počítače*** ("the computers are ideal for them"; lit. "for them are ideal computers").
5. Prepositional phrase instead of an adverb: *o patnáct procent* ***větší obrat*** (lit. "of fifteen percent higher turn-over"). Such cases are not solvable in general; the PP attachment is a complex task that requires a statistical analysis of lexical elements involved. However, some portion with predefined prepositions could be captured.
6. Instrumental case noun instead of an adverb: *archeology* ***zkoumaných pramenů*** ("resources investigated by archeologists"). Perhaps could be solved because adjectives derived from verbs can be recognized after their tags.
7. Deletions: *čím* ***větší*** *~~je~~* ***odchylka****, tím víc čeká…* ("the more the deviation ~~is~~, the more…). Insolvable. The verb "is" is deleted and all its dependents shall hang on its governor (according to the definition of AL). If the verb was present it would mark a nominal predicate and this error could be solved as no. 4.
8. Non-projective constructions: ***tématické zájezdy****, například odborníků ze zemědělství* ("theme tours, e.g. tours of agriculture experts"; the second "tours" is not repeated in Czech original and whole "e.g."-phrase is considered to form an apposition together with the adjective "theme"). Insolvable. If the construction was projective, the phrase would not be recognized but no error would arise.
9. Errors in annotation. I considered some cases where the RE proposed a dependency different from that annotated in the "gold standard" to be mistake of the annotators, not an error done by the RE. Such cases are not too frequent; though they occur for (nearly) all RE's I tested. I will not mention them explicitly when describing the remaining RE's.

4.3. Adverb – adjective AP's

The following RE, called DA, recognizes adjective phrases consisting of an adjective preceded by an adverb.

```
"<t>D"
"<t>A"
```

This is the most problematic pattern I tested. 1800 good and 622 bad phrases were recognized, giving an accuracy of only 74.3 %. The problem is that adverbs often modify a verb and the uncertainty between adjectival and verbal attachment can be compared to the PP attachment problem.

One may ask whether listing adverbs that usually attach to a verb rather than adjective can solve the problem. I modified the procedure slightly to report adverbs that contributed to at least five bad phrases. Then I restricted the RE to phrases with these naughty adverbs. The result showed that even these adverbs were correct in more cases than incorrect: 856 good, 449 bad, accuracy = 65.6 %. Even restricting to adverbs that caused ten or more errors (*tedy* / thus, *ovšem* / indeed, *totiž* / namely, *včetně* / including, *ještě* / still, *tak* / so, *rovněž* / likewise, *především* / above all, *jak* / how, *více* / more) did not lead to more bad than good phrases: 273 good, 268 bad, accuracy = 50.5 %. So there is nothing like verb liking adverbs, once the right context is adjective.

Although using of this RE is controversial I included it in final tests. With one change however: I restricted it to the adverbs with tag `Dg` (adverbs derived from adjectives). The accuracy of such phrases is 86.8 % but, naturally, less good phrases are found (930 good, 141 bad).

4.4. Noun – genitive noun NP's

One common way of forming noun phrases in Czech (and other Slavic languages — see Nenadić (1998) for Serbo-Croatian) is accompanying an existing NP with another NP in genitive case. The genitive NP has a similar function as a

PP with the preposition *of* in English. So, for instance *nový prezidentský mandát Václava Havla* means "new presidential term of Václav Havel". The whole NP consists of two NP's, *nový prezidentský mandát* and *Václava Havla*. The members of each of these NP's agree in gender, number and case. However, the second NP is frozen in genitive regardless in which case the first NP is.

Theoretically, an unlimited number of genitives can be recursively cumulated this way — consider phrases like "the break-down of the new car of the father of a friend of mine". We need to recognize the rightmost pair of genitives first to get the whole tree correct. So the RE recognizing two consecutive genitives must make sure that there are no more genitives to the right. The (obvious) base RE called NgNg is as follows:

```
"<t>N...2"
"<t>N...2"
```

After the procedure creates one RE from the above pieces, we must add the following negative look-ahead condition. It assures that the next word is not a noun in genitive:

```
"(?!.*?\\n.*?<t>N...2.*?\\n)"
```

When all genitive pairs are found and collapsed, the noun-in-any-case – noun-genitive pairs can collapse. This is done by the RE called NNg:[5]

```
"<t>N"
"<t>N...2"
```

Both RE's together found 13925 good and 1233 bad phrases, which gives an accuracy of 91.9 %. Errors often arose in sequences of genitives wherever the structure was not purely right branching: *auto Michaela Douglase* means "the car of Michael Douglas" in English but would be analyzed as "the car of Michael of Douglas". We faced a similar problem with noun coordinations but here it is much more difficult to find a solution since the different interpretations are based rather semantically.

4.5. Prepositional phrases[6]

The following RE, called RN, combines prepositions with nouns whose case is compatible with the respective preposition:

```
"<t>R...(\\d)"
"<t>N...\\1"
```

This RE is quite successful. The relatively small number of errors it does would further decrease if there was a more fine grained model of coordinations and other phrases mentioned above. An analysis of compound expressions with prepositional function (such as *na rozdíl od*, "as opposed to") would help as well.

The RN RE found 21294 good and 965 bad prepositional phrases, which gives an accuracy of 95.7 %.

4.6. All together now!

Finally I tested all described RE's in the following order (note that NaN was applied twice):

```
DA-AaA-AN-NaN-NgNg-NNg-NaN-RN
```

The test was done on 19126 sentences of the Prague Dependency Treebank; this portion contains 346719 words (tokens). The bracketing done by the RE preprocessor proposed 69262 correct dependencies and 4424 false ones while leaving 273033 tokens unresolved. So the recall is 20.0 %, which means that the preprocessor is able to (correctly) help in one fifth of cases. The precision of the preprocessor is 94.0 %; only 1.3 % of all dependencies are errors that the parser cannot repair. In other words, if the parser was otherwise prefect, it could reach an accuracy of 98.7 %. This is to be compared to the accuracy of current parsers available for Czech (Collins 80 %, all others under 70 %; see Hajič et al., 1998, and Collins et al., 1999, for details).

The above results are for data where morphological tags were assigned manually. I find such evaluation important because it shows the power of the RE's without biasing it by errors of other components. However, an application will hardly have manually annotated data available, so I'm adding two other statistics. First, the RE set was run on tags and lemmas assigned by a statistical tagger (Hajič and Hladká, 1998). Every `<t>` in RE's was automatically replaced by `<MDt[^>]*>`; the `<l>` markups were treated similarly. The resulting machine proposed 60093 good dependencies and

[5] Both RE's could be combined into one (NNg with the look-ahead condition).

[6] Some more statistics of different prepositional phrase patterns in Czech can be found in (Petkevič, 1999). Note however that they were collected from a different corpus.

5001 bad dependencies, 281625 remained unresolved; the precision was 92.3 %, the recall 17.3 %. The decrease in both the precision and the recall can be explained by the errors the tagger does.

The last test used ambiguous morphological input. Every `<t>` in RE's was automatically replaced by `<MMt>`; the `<l>` markups were treated similarly. The resulting machine proposed 69566 good dependencies and 8146 bad dependencies, 269007 remained unresolved; the precision was 89.5 %, the recall 20.1 %. The preprocessor had all morphological hypotheses available[7], so the recall slightly increased. But occasionally it combined two compatible hypotheses that occurred at neighboring positions but were wrong. That explains the drop in precision.

5. Conclusion

I showed how a set of regular expressions can be very easily and rapidly built and applied to find chunks of syntactic sentence structure. I proposed the RE's be used as a preprocessor before a full parser. This may help both the parser's precision and its speed, since the input of the parser will be simpler. In Section 2 I proposed two ways of combining the preprocessor with the parser but the evaluation of these methods is left for near future work.

The work I presented is not "pure" and systematic from the theoretical point of view — I have shown some problems that cannot be solved easily in my framework. But at the same time I showed that the problematic constructions appear quite rarely in real data, which makes the approach practical and justifies its use. However, it is impossible to make the final judgment before the RE preprocessor is evaluated together with a parser and its real contribution is seen.

6. Acknowledgements

This research is being supported by the Ministry of Education of the Czech Republic project No. LN00A063 (Center for Computational Linguistics).

The data (PDT) would not be available unless the grant No. 405/96/K214, of the Grant Agency of the Czech Republic, enabled work on the treebank design. (Granted to the Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Prague.)

7. References

Alena Böhmová, Jan Hajič, Eva Hajičová, Barbora Hladká (in press). The Prague Dependency Treebank: Three-Level Annotation Scenario. In: Anne Abeillé (Ed.): *Treebanks: Building and Using Syntactically Annotated Corpora.* Kluwer Academic Publishers, Dordrecht, The Netherlands. See also http://ufal.mff.cuni.cz/pdt/.

Michael Collins, Jan Hajič, Eric Brill, Lance Ramshaw, Christoph Tillmann (1999). A Statistical Parser of Czech. In: *Proceedings of the 37th Meeting of the ACL*, pp. 505–512, University of Maryland, College Park, Maryland.

Jan Hajič, Eric Brill, Michael Collins, Barbora Hladká, Douglas Jones, Cynthia Kuo, Lance Ramshaw, Oren Schwartz, Christoph Tillmann, Daniel Zeman (1998). *Core Natural Language Processing Technology Applicable to Multiple Languages. The Workshop 98 Final Report*. At: http://www.clsp.jhu.edu/ws98/ projects/nlp/report/. Johns Hopkins University, Baltimore, Maryland. Shortened version in PBML vol. 70, pp. 73–82.

Jan Hajič, Barbora Hladká (1998). Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In: *Proceedings of the 36th Meeting of the ACL and COLING'98*, pp. 483–490. Université de Montréal, Montréal, Québec.

Goran Nenadić, Duško Vitas (1998). Using Local Grammars for Agreement Modeling in Highly Inflective Languages. In: Petr Sojka, Václav Matoušek, Karel Pala, Ivan Kopeček (Eds.): *Text, Speech and Dialogue, Proceedings of the 1st International Workshop,* pp. 91–96, Brno, Czechia.

Goran Nenadić (2000). Local Grammars and Parsing Coordination of Nouns in Serbo-Croatian. In: Petr Sojka, Ivan Kopeček, Karel Pala (Eds.): *Text, Speech and Dialogue, Proceedings of the 3rd International Workshop,* pp. 57–62, Springer LNAI 1902, Brno, Czechia.

Vladimír Petkevič (1999). Czech Translation of G. Orwell's '1984': Morphology and Syntactic Patterns in the Corpus. In: Václav Matoušek, Pavel Mautner, Jana Ocelíková, Petr Sojka (Eds.): *Text, Speech and Dialogue, Proceedings of the 3rd International Workshop,* pp. 77–82, Springer LNAI 1692, Mariánské Lázně, Czechia.

Larry Wall, Tom Christiansen, Randal Schwartz (1996). *Programming Perl.* O'Reilly. ISBN 1-56592-149-6. Or see http://www.perl.org/.

Daniel Zeman (1998). *A Statistical Approach to Parsing of Czech.* In: Prague Bulletin of Mathematical Linguistics, vol. 69, pp. 29–37. Univerzita Karlova, Praha, Czechia.

[7] There is a potential to use the preprocessor as a morphological disambiguator as well. It has to be investigated separately.