Improving Corpus Annotation Productivity: a Method and Experiment with Interactive Tagging

Atro Voutilainen

Department of Modern Languages
University of Helsinki
first.last@helsinki.fi

Abstract

Corpus linguistic and language technological research needs empirical corpus data with nearly correct annotation and high volume to enable advances in language modelling and theorising. Recent work on improving corpus annotation accuracy presents semiautomatic methods to correct some of the analysis errors in available annotated corpora, while leaving the remaining errors undetected in the annotated corpus. We review recent advances in linguistics-based partial tagging and parsing, and regard the achieved analysis performance as sufficient for reconsidering a previously proposed method: combining nearly correct but partial automatic analysis with a minimal amount of human postediting (disambiguation) to achieve nearly correct corpus annotation accuracy at a competitive annotation speed. We report a pilot experiment with morphological (part-of-speech) annotation using a partial linguistic tagger of a kind previously reported with a very attractive precision-recall ratio, and observe that a desired level of annotation accuracy can be reached by using human disambiguation for less than 10% of the words in the corpus.

Keywords: corpus annotation, interactive tagging, treebanking

1. Introduction

Tagged and parsed text corpora are needed for corpus linguistics and for building and testing language models for use in automatic language analysis. To maximise representativeness and reliability, attention needs to be paid to size and annotation accuracy (including annotation guidelines and annotation consistency evaluations). Annotating and documenting a 20-40 thousand sentence treebank can take several years of work (Abeille, 2003; Hwa et al., 2005). For corpus linguistic studies with a mixed syntactic and lexical focus, much larger annotated corpora are probably needed, which makes fully manual corpus annotation and treebanking impractical. On the other hand, the consistency and accuracy of the annotation effort can remain somewhat inadequate for corpus linguistic studies and for enabling new advances in development of sufficiently accurate statistical language models for automatic corpus annotation (tagging and parsing) (Manning, 2011).

During the past twenty-five years, substantial progress has been made on linguistics-based language modelling and parsing, most successfully in the Constraint Grammar (CG) framework (Karlsson et al., 1995; Samuelsson and Voutilainen, 1997; Tapanainen and Järvinen, 1997; Bick, 2000). A CG can be designed to provide a partial but nearly error-free analysis to its input: some of the more difficult ambiguity is passed on for later processing in the pipeline, but the analysis decisions made by this kind of CG are sufficiently reliable for the intended application. This makes an old but largely ignored annotation method worth reconsidering: interactive annotation. We (re)propose a method to enable high-accuracy treebank annotation with competitive annotation speed:

- most of the annotation is done automatically with a partial, reductionistic parser that uses constraint-based manually-built language models with sufficient reliability (recall) for the application;
- the human annotator resolves only pending ambiguities, one per analysis unit (sentence);
- based on the additional disambiguation by the human annotator, the parser continues the disambiguation towards a complete sentence analysis.

Next, we look at previous work on methods for more effective corpus annotation. In section 3, we outline our framework and method. Section 4 reports on a small-scale empirical evaluation with morphology (word-class) annotation. The paper ends with a discussion and a look at future work.

2. Previous Work

In an early project to annotate the Brown University Corpus of American English, a reductionistic tagger with linguistic grammars (TAGGIT) was used in order to speed up the annotation process. The method by Greene and Rubin (1971) is largely similar to ours: a partial tagger annotates and disambiguates the majority (over 70% of the words) of the corpus; the human expert disambiguates those words not disambiguated by TAGGIT. A reported weakness in the TAGGIT analyser was that several percentage points of the disambiguations it made were erroneous (i.e. the correct word-class tag was discarded). As a result, manual revision and postediting was needed for the whole corpus, which probably severely compromised the intended work savings.

Recently, research on identifying and correcting misanalysis in complete output (e.g. parse trees) provided by a statistical tagger or parser has been reported (Dickinson and Meurers, 2003; Loftsson, 2009; Dickinson and Smith, 2011; Manning, 2011). Hand-coded or automatically generated heuristics are executed on the parsed corpus to identify utterances whose analysis likely need a correction by a human posteditor. The main rationale of this approach is that at least some of the misanalyses (Dickinson and Smith estimate: 50% of misanalyses) can be identified and corrected by manually postediting a smaller part of the parsed corpus (Dickinson and Smith: 20% of the corpus). These methods enable partial correction of the corpus with reasonable efficiency, but a substantial amount of annotation errors remains unlocated and uncorrected.

3. Framework and Method

Our method is based on use of certain results and properties of a linguistic framework for surface-syntactic finite state tagging and parsing (Koskenniemi et al., 1992; Karlsson et al., 1995; Tapanainen and Järvinen, 1997) known as Finite State Intersection Grammar, Constraint Grammar and Functional Dependency Grammar:

- linguistic analysis tasks and representations (e.g. tag sets) can be specified and documented in sufficient detail and clarity to enable trained linguists to manually apply the representations with nearly 100% uniformity (Voutilainen and Järvinen, 1995; Voutilainen, 1999b; Voutilainen and Purtonen, 2011); high specifiability of linguistic representations is a necessary prerequisite for aiming at high-accuracy corpus annotation.

- the language models applied by the tagging/parsing software are based on the linguist's abstractions and corpus observations; frequencies and statistics are not encoded in the resulting lexicons and grammars. (However, linguistic models and analysers can be combined with statistical ones to create hybrid models and analysers).

- aiming at a complete correct analysis for all inputs is generally regarded as unrealistic; hence the parsing grammarian can control the precision/recall tradeoff of the resulting parser by the level of detail, in which context conditions are formulated for syntactic analysis operations (REMOVE an illegitimate analysis; SELECT a correct analysis by removing alternatives; ADD a named dependency relation between specified words in the sentence). Also, the grammar can be organised into ordered subgrammars: heuristic subgrammars with a higher error margin can be written to resolve ambiguities left pending by the initially applied 'strict' subgrammars.

- rules in a parsing grammar can feed each other: a constraint (or many) can disambiguate a part of a sentence, as a result of which that sentence fragment can serve as a sufficiently unambiguous context condition for another constraint to enable disambiguation (or structure building) for another part of the sentence.

- with carefully built and tested linguistic rules, very attractive trade-offs between precision and recall can be achieved. In a comparative evaluation (Samuelsson and Voutilainen, 1997) it was showed that with a common tagset and analysis task, a state-of-the-art statistical tagger mispredicted the wordclass 9 to 28 times more frequently than the linguistic tagger, depending on the (equal) amount of ambiguity both systems were allowed to leave unresolved.

- the same formalism and methodology can be used for introducing and resolving several layers of linguistic representation and analysis (e.g. morphology, word-class disambiguation, phrase chunking, shallow function syntax, dependency relation analysis, etc; cf. (Didriksen, 2011).

We propose the following method:

1. apply a large-coverage lexical analyser to the corpus to introduce a morphological analysis to each word in the corpus (and alternative analyses to ambiguous words). (In a corpus annotation effort, the morphological analyser's lexicon, usually based on publicly available lexical resources, is updated with lexis in the corpus for complete coverage.)

2. apply a partial but reliable disambiguator on the lexically analysed corpus (reliability relative to agreed annotation accuracy goals of the project). As a result, there is no need to reconsider (postedit) analyses by the machine.

3. for each sentence with a pending ambiguity, let a human expert resolve an ambiguity (e.g. using custom-built editor macros to enable examination and disambiguation with a simple click). For higher human annotation consistency and accuracy, the analysis can be made following the double-blind method (as described in (Voutilainen, 1999a) given sufficient resources.

4. repeat (ii) to enable further disambiguation based on the piece of information provided by the expert, and then (iii), until the sentence (or corpus) is fully disambiguated.

5. add the next level of linguistic representation to the disambiguated corpus, and follow (ii) through (v) until desired level of analysis is achieved.

4. Tagger

An EngCG-style modular tagging system with hand-coded language models for tokenisation, morphological analysis and contextual disambiguation was used in the experiment. The morphological distinctions are based on Quirk et al. (1985), and are somewhat more fine-grained than what is commonly used in part-of-speech taggers:

- non-inflected verb forms are given four separate readings to distinguish between present tense, subjunctive, infinitive and imperative readings;

```
"<state>"
     "state" N_NOM_SG
     "state" V_PRES_-SG3
     "state" V_IMP
     "state" V_SUBJUNCTIVE
     "state" V_INF
```

- participial forms (e.g. ‘giving’ and ‘given’) are given separate nominal and verbal analyses (while the original EngCG tagger introduced these distinctions for participial forms at a later stage of syntactic analysis);

```
"<following>"
     "following" Ing_A_ABS
     "follow" V_ING
     "following" Nom_ING
"<thought>"
     "thought" N_NOM_SG
     "thought" Nom_EN
     "think" V_PAST
     "think" V_EN
```

- a distinction is systematically made between prepositions and subordinating conjunctions (while taggers using only local context that is often insufficient to resolve these distinctions tend to collapse this distinction under a single tag;

```
"<as>"
     "as" ADV
     "as" CS
     "as" PREP
```

- a distinction is made between pronoun and determiner analyses for closed-class words like ‘that’, ‘those’, ‘what’, ‘some’;

```
"<those>"
     "that" DET_CENTRAL_DEM_PL
     "that" PRON_DEM_PL
"<some>"
     "some" ADV
     "some" Quant_PRON_SG/PL
     "some" Quant_DET_CENTRAL_SG/PL
"<many>"
     "many" Quant_DET_POST_ABS_PL
     "many" Quant_DET_PRE_ABS_SG
     "many" Quant_PRON_ABS_PL
"<that>"
     "that" ADV
     "that" CS
     "that" DET_CENTRAL_DEM_SG
     "that" PRON_DEM_SG
     "that" Rel_PRON_SG/PL
```

- a distinction is made between interrogative and relative pronouns;

```
"<which>"
     "which" Interr_PRON_WH_NOM_SG/PL
     "which" Rel_PRON_WH_NOM_SG/PL
     "which" DET_CENTRAL_WH_SG/PL
```

- a distinction is made between the preposition and infinitive marker uses of ‘to’ and the multiword ‘in_order_to’:

```
"<to>"
     "to" INFMARK
     "to" PREP
```

Initially, a tokeniser and a morphological analyser with a large lexicon and EngCG-style morphology is used for introducing morphological analyses and ambiguities. Here is a small sample analysis; each "cohort" consists of the word-form and one or more alternative analyses (base form and morphological tag), each on an indented line:

```
"<There>"
     "there" ADV
     "there" Ex_ADV
"<,>"
     "," PUNCT
"<in>"
     "in" ADV
     "in" PREP
"<the>"
     "the" Def_DET_CENTRAL_ART_SG/PL
"<Free>"
     "free" A_ABS
     "free" V_PRES_-SG3
     "free" V_IMP
     "free" V_SUBJUNCTIVE
     "free" V_INF
"<Territory>"
     "territory" N_NOM_SG
"<,>"
     "," PUNCT
"<they>"
     "they" PRON_PERS_NOM_PL3
"<fought>"
     "fought" Nom_EN
     "fight" V_PAST
     "fight" V_EN
"<in>"
     "in" ADV
     "in" PREP
"<the>"
     "the" Def_DET_CENTRAL_ART_SG/PL
"<civil>"
     "civil" A_ABS
"<war>"
     "war" N_NOM_SG
     "war" V_PRES_-SG3
     "war" V_IMP
     "war" V_SUBJUNCTIVE
     "war" V_INF
"<against>"
     "against" PREP
"<the>"
     "the" Def_DET_CENTRAL_ART_SG/PL
"<Whites>"
     "white" N_NOM_PL
     "white" V_PRES_SG3
"<.>"
     "." PUNCT
```

The disambiguation grammar consists of one or more sequential subgrammars: constraints in the first subgrammar are applied to the input sentence until no more disambiguation is done; the following subgrammar is also applied to resolve remaining ambiguity. The first subgrammars usually contain reliable constraints; heuristic constraints usually based on somewhat simplified linguistic generalisations are listed in the last subgrammar(s). A mature grammar usually contains 3-5 subgrammars and a few thousand constraints.

A constraint removes alternative analyses if its contextual tests succeed. Linguistically, a constraint typically expresses a syntactic generalisation but in a partial manner, referring to syntactic and lexical categories (tags and their sequences, as well as base forms or sets thereof). Here we give some simple constraints that can be applied to the above sample sentence. – The following constraint removes readings listed in the set 'verb' from ambiguous cohorts if somewhere to the left, there is a cohort with an unambiguous determiner reading, and there is no intervening cohort with a tag belonging to the set 'nphead':

```
REMOVE verb
      (**-1C determiner BARRIER nphead) ;
```

"Remove "

The following constraint selects a finite verb reading as correct (i.e. discards all alternative morphological readings) if the sentence contains no other words with a finite verb reading:

```
SELECT finite-verb
      (NOT *-1 finite-verb)
      (*1 fullstop BARRIER finite-verb) ;
```

The third sample constraint also uses lexical criteria. The adverb reading of 'in' is deleted if the previous cohort does not contain baseforms in the set 'word-with-IN-as-particle' (e.g. "put" and "give") and if the following word contains tags appropriate for premodifiers or heads of a noun phrase:

```
REMOVE ("in" ADV)
      (NOT -1 word-with-IN-as-particle
              OR conj-coord)
      (1C premodifier OR nphead) ;
```

Let us look at how the constraints apply to the sample sentence. The first constraint disambiguates three words: 'Free', 'war' and 'Whites'; this enables the second constraint to disambiguate 'fought'. The last constraint disambiguates all instances of 'in'. Only 'There' remains ambiguous:

```
"<There>"
      "there" ADV
      "there" Ex_ADV
"<,>"
      "," PUNCT
"<in>"
      "in" PREP
"<the>"
      "the" Def_DET_CENTRAL_ART_SG/PL
"<Free>"
      "free" A_ABS
"<Territory>"
      "territory" N_NOM_SG
"<,>"
      "," PUNCT
"<they>"
      "they" PRON_PERS_NOM_PL3
"<fought>"
      "fight" V_PAST
"<in>"
      "in" PREP
"<the>"
      "the" Def_DET_CENTRAL_ART_SG/PL
"<civil>"
      "civil" A_ABS
"<war>"
      "war" N_NOM_SG
"<against>"
      "against" PREP
"<the>"
      "the" Def_DET_CENTRAL_ART_SG/PL
"<Whites>"
      "white" N_NOM_PL
"<.>"
      "." PUNCT
```

In the pilot experiment, a part of the morphological ambiguity was resolved using a nontrivial morphological disambiguation grammar with linguist-written non-heuristic constraints (corresponding to the initial non-heuristic subgrammars in EngCG) tested and refined with tagged and untagged corpora of Present-Day Standard English).

5. Pilot Experiment

We report a small experiment with human-assisted corpus annotation at the level of morphology (word class and inflectional tags), using an extract from the English Wikipedia as our test corpus. We look at the human workload (number of human classifications) needed to annotate a corpus with a near-100% accuracy.

The test corpus is a 4288-word extract from a Wikipedia article "Anarchism".

After morphological analysis, the corpus received 7803 analyses (1.8 analyses per word on an average). After applying the disambiguation grammar, 377 words (8.8% of all words) remained morphologically ambiguous. The human expert disambiguated the first ambiguity in each ambiguous sentence (using a purpose-built emacs macro on a click-per-choice basis), and the disambiguator was applied to the corpus to enable further automatic disambiguation. For instance, by selecting an Determiner analysis as correct in a Pronoun/Determiner ambiguity, the machine equipped with the disambiguation grammar was able to resolve a Noun/Verb ambiguity in the right-hand context in favour of the Noun reading when looking for a suitable head for the human-selected Determiner reading.

The human-machine disambiguator loop was repeated six times for completed disambiguation; the human disambiguation was (disappointingly) needed in as many as 369 cases out of the total 377.

6. Discussion and Future Work

Recent studies have shown that postediting annotated corpora for sufficient tagging correctness with support from automatic error detection heuristics is somewhat work-intensive, and leaves a substantial part of all annotation errors in the corpus undetected.

Based on certain important characteristics of and empirical results from a linguistic surface-syntactic analysis model introduced in the early 1990s, we have outlined an interactive annotation method to combine the benefits of nearly 100% reliable automatic reductionistic annotation and disambiguation with sparing use of human disambiguation to provide means for very accurate and efficient corpus annotation.

We have reported a pilot experiment with word-class annotation of English, which indicates that over 90% of the annotation can be done reliably by a linguistic tagger (no need for human revision), while the human expert (or experts in a double-blind + negotiation mode) can resolve remaining nontrivial cases effectively and consistently, by using appropriate tools and annotation guidelines.

The obtainable annotation accuracy (close to 100% at word-class level) and annotation speed (human analysis needed for less than 10% of the corpus) is a potential alternative to existing methods.

Additional annotation work and experimentation with other analysis levels and languages as well as linguistic resources and toolkits enable a more thorough development and evaluation of the proposed approach.

The results of such continued work can be expected to yield competitively-sized annotated corpora with very high annotation quality for the research community, which enables experimentation with improved statistical and machine learning methods for higher-accuracy automatic analysis of running text.

Acknowledgements

The research has been supported by FIN-CLARIN, META-NORD and the Academy of Finland. The author is grateful to three anonymous reviewers for their constructive comments, and to Kristiina Muhonen, Tanja Purtonen and Sam Hardwick for assistance with the experiments and preparing the paper.

7. References

Anne Abeille, editor. 2003. *Treebanks: Building and using syntactically annoted corpora*. Kluwer Academic Publishers.

Arto Anttila. 1995. How to recognise subjects in English. In Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, editors, *Constraint Grammar: A Language-Independent System for Parsing Running Text*. Mouton de Gruyter, Berlin and New York.

Eckhard Bick. 2000. *The Parsing System "Palavras"*. Aarhus University Press, Aarhus.

Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*. ACL.

Markus Dickinson and Amber Smith. 2011. Detecting dependency parse errors with minimal resources. In *Proceedings of the 12th International Conference on Parsing Technologies*. ACL.

Tino Didriksen. 2011. Constraint Grammar Manual: 3rd version of the CG formalism variant. GrammarSoft ApS. http://beta.visl.sdu.dk/cg3/vislcg3.pdf.

Barbara B. Greene and Gerald M. Rubin. 1971. Automated grammatical tagging of English. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*.

Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, editors. 1995. *Constraint Grammar: A Language-Independent System for Parsing Running Text*. Number 4 in Natural Language Processing. Mouton de Gruyter, Berlin and New York. ISBN 3-11-014179-5.

Kimmo Koskenniemi, Pasi Tapanainen, and Atro Voutilainen. 1992. Compiling and using finite-state syntactic rules. In *Proceedings of the 15th International Conference on Computational Linguistics*, volume I, pages 156–162, Nantes, France. ICCL.

Hrafn Loftsson. 2009. Correcting a POS-Tagged Corpus using three complementary methods. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. ACL.

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing, 12th International Conference, CICLing 2011, Proceedings, Part I*, Lecture Notes in Computer Science 6608. Springer.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.

Christer Samuelsson and Atro Voutilainen. 1997. Comparing a linguistic and a stochastic tagger. In *Proceedings of the 35th Annual Meetng of the Association for Computational Linguistics and the Eighth Conference of the European Chapter of the Association for Computational Linguistics*. ACL.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*. ACL.

Atro Voutilainen and Timo Järvinen. 1995. Specifying a shallow grammatical representation for parsing purposes. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics (EACL 1995)*. ACL.

Atro Voutilainen and Tanja Purtonen. 2011. A double-blind experiment on interannotator agreement: The case of dependency syntax and Finnish. In *NODALIDA 2011 Conference Proceedings*, pages 319–322.

Atro Voutilainen. 1993. Nptool, a detector of English noun phrases. In *Proceedings of the Workshop on Very*

Large Corpora: Academic and Industrial Perspectives (WVLC). ACL.

Atro Voutilainen. 1999a. An experiment on the upper bound of interjudge agreement: the case of tagging. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999)*, pages 204–208. ACL.

Atro Voutilainen. 1999b. Hand-crafted rules. In Hans van Halteren, editor, *Syntactic Wordclass Tagging*. Kluwer Academic Publishers, Dortrecht, Boston and London.

Appendix: Morphological Tags

The tag palette (106 tags) used by the tagger closely resembles that documented in (Karlsson et al., 1995). The main difference is that for participial ING and EN forms (e.g. “giving”, “given”), the word-class ambiguity between adjective, noun and verb readings is spelled out when appropriate, while the original ENGCG morphology did not spell out the word-class ambiguity for these forms, postponing their resolution for later stages of analysis, e.g. chunking (Voutilainen, 1993) or dependency function assignment (Anttila, 1995).

```
A_ABS A_CMP A_SUP ADV ADV_ABS
ADV_CMP ADV_SUP ADV_WH CC CS
Def_DET_CENTRAL_ART_SG/PL
DET_CENTRAL_DEM_PL DET_CENTRAL_DEM_SG
DET_CENTRAL_SG DET_CENTRAL_WH_GEN_SG/PL
DET_CENTRAL_WH_SG/PL
DET_POST_SG DET_POST_SG/PL
DET_PRE_SG/PL DET_PRE_WH_SG/PL
EN=Nom EN=V Ex_ADV
Genord_DET_POST_SG/PL
Indef_DET_CENTRAL_ART_SG
INFMARK Ing_A_ABS ING=Nom ING=V
Interr_PRON_WH_GEN_SG/PL
Interr_PRON_WH_NOM_SG/PL N NEG-PART
N_GEN_PL N_GEN_SG N_NOM_PL N_NOM_SG
N_NOM_SG/PL NUM_CARD NUM_ORD PREP
PRON_ACC_SG3 PRON_DEM_PL PRON_DEM_SG
PRON_GEN_SG3 PRON_NOM_PL
PRON_NOM_SG PRON_NOM_SG3 PRON_NOM_SG/PL
PRON_PERS_ACC_PL1 PRON_PERS_ACC_PL3
PRON_PERS_GEN_PL1 PRON_PERS_GEN_PL3
PRON_PERS_MASC_ACC_SG3
PRON_PERS_MASC_GEN_SG3
PRON_PERS_MASC_NOM_SG3
PRON_PERS_NOM_PL3 PRON_PERS_NOM_SG1
PRON_RECIPR PRON_WH_SG/PL
Proper_N_GEN_SG Proper_N_NOM_SG
Proper_N_NOM_SG/PL PUNCT
Quant_DET_CENTRAL_SG
Quant_DET_CENTRAL_SG/PL
Quant_DET_POST_ABS_PL
Quant_DET_POST_ABS_SG
Quant_DET_POST_CMP_PL
Quant_DET_POST_CMP_SG
Quant_DET_POST_PL
Quant_DET_POST_SUP_PL
Quant_DET_POST_SUP_SG
Quant_DET_PRE_ABS_SG Quant_DET_PRE_PL
Quant_DET_PRE_SG/PL
Quant_N_NOM_SG/PL Quant_PRON_ABS_PL
Quant_PRON_ABS_SG
Quant_PRON_CMP_PL Quant_PRON_CMP_SG
Quant_PRON_CMP_SG/PL
Quant_PRON_NOM_PL Quant_PRON_NOM_SG
Quant_PRON_NOM_SG/PL
Quant_PRON_SG Quant_PRON_SG/PL
Quant_PRON_SUP_PL
Quant_PRON_SUP_SG Refl_PRON_PERS_MASC_SG3
Refl_PRON_PERS_PL3
Refl_PRON_SG3 Rel_PRON_SG/PL
Rel_PRON_WH Rel_PRON_WH_GEN_SG/PL
Rel_PRON_WH_NOM_SG/PL V_AUXMOD
V_IMP V_INF V_PAST V_PAST_PL V_PAST_SG1,3
V_PRES_-SG1,3 V_PRES_SG3 V_PRES_-SG3
V_SUBJUNCTIVE
```