Hybrid processing for grammar and style checking

Berthold Crysmann[♣], Nuria Bertomeu[♠], Peter Adolphs[♡], Dan Flickinger[♢], Tina Klüwer[♣♡]

[♣] Universität Bonn, Poppeldorfer Allee 47, D-53115 Bonn, {bcr,tkl}@ifk.uni-bonn.de
[♠] Zentrum für Allgemeine Sprachwissenschaft, Berlin, nuria.bertomeu@dfki.de
[♡] DFKI GmbH, Berlin, {peter.adolphs,kluewer}@dfki.de
[♢] CSLI, Stanford University, danf@csli.stanford.edu

Abstract

This paper presents an implemented hybrid approach to grammar and style checking, combining an industrial pattern-based grammar and style checker with bi-directional, large-scale HPSG grammars for German and English. Under this approach, deep processing is applied selectively based on the error hypotheses of a shallow system. We have conducted a comparative evaluation of the two components, supporting an integration scenario where the shallow system is best used for error detection, whereas the HPSG grammars add error correction for both grammar and controlled language style errors.

1 Introduction

With the enormous amount of multilingual technical documentation produced by companies nowadays grammar and controlled language checking (henceforth: style checking) is becoming an application highly in demand. It is not only a helpful tool for authors, but also facilitates the translation of documents into foreign languages. Through the use of controlled language by the authors, documents can be automatically translated more successfully than with the use of free language. Style checking should make authors aware of the constructions which should not be used, as well as aiding in reformulating them. This can save a lot of translation costs for companies producing large amounts of mulitilingual documentation. Another application of grammar and style checking is the development of tutorial systems for learning a foreign language, as well as any kind of authoring system for non-native speakers.

Previous approaches to grammar and style checking can be divided into those based on finite state methods and those based on linguistically motivated grammars. To the former group belong e.g. the systems FLAG (Bredenkamp et al., 2000a; Bredenkamp et al., 2000b) and MultiLint (Haller, 1996; Schmidt-Wigger, 1998). The basic approach taken by such systems is the description of error patterns through finite state automata. The automata access the textual input enriched with annotations from shallow linguistic analysis components, such as part-of-speech tagging, morphology and chunking. In FLAG, for instance, the annotation delivered by the shallow components is integrated into a complex feature structure. Rules are defined as finite state automata over feature structures. The great advantages of such systems are their robustness and efficient processing, which make them highly suitable for real-life grammar and style checking applications. However, since shallow modules usually cannot provide a full syntactic analysis, the coverage of these systems is limited to error types not requiring a broader (non-local) syntactic context for their detection. Therefore their precision in the recognition of non-local errors is not satisfactory.

Another short-coming of most shallow approaches to grammar checking is that they typically do not provide error correction: owing to the absence of an integrated target grammar, generation of repairs cannot take the syntactic context into account: as a result, some of the repairs suggested by shallow systems are not globally well-formed.

Grammar-based error checking constitutes the other main strand in language checking technology. These systems are typically equipped with a model of target well-formedness. The main problem, when applied to the task of error checking is that the sentences that are the focus of a grammar checker are ideally outside the scope of the grammar. To address this problem, grammar-based checkers typically employ robustness techniques (Ravin, 1988; Jensen et al., 1993; Douglas, 1995; Menzel, 1998; Heinecke et al., 1998). The addition of robustness features, while inevitable for a grammar-based approach, has the disadvantage of considerably slowing down runtime performance. Another issue with purely grammar-based checking is related to the scarce distribution of actual errors: thus, most effort is spent on the processing of perfectly impeccable utterances. Finally, since coverage of real-world grammars is never perfect, these system also have difficulty to distinguish be-

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 153–160
Manchester, August 2008

tween extragrammatical and truly ungrammatical sentences. Conversely, since grammars often overgenerate, a successful parse does not guarantee wellformedness either.

One of the two major robustness techniques used in the context of grammar-based language checking are constraint relaxation (see e.g. (Douglas, 1995; Menzel, 1998; Heinecke et al., 1998)), which is typically realised by means of modifications to the parser (e.g. relaxation levels, robust unification). An alternative approach is error anticipation where errors are explicitly modelled by means of grammar rules, so-called MAL-rules (McCoy et al., 1996). This approach has already been investigated with an HPSG grammar, the ERG (Copestake and Flickinger, 2000), in the scenario of a tutorial system for language learning by (Bender et al., 2004). We will follow this approach in the part of our hybrid system based on deep processing.

Finite state methods and linguistically motivated grammars are not only compatible, but also complementary. Shallow methods are robust and efficient, while deep processing based on grammars provides high precision and detail. With the focussed application of deep analysis in finite state based grammar and style checking systems, both coverage and precision can be improved, while the performance remains acceptable for real-world applications. The combination of shallow and deep components, hybrid processing, has already been investigated in several modular architectures, such as GATE (Gaizauskas et al., 1996), Whiteboard (Crysmann et al., 2002) and Heart-of-Gold (Callmeier et al., 2004). Moreover, the improvement in efficiency and robustness in deep processing together with methods for its efficient application makes the employment of deep processing in real-world applications quite feasible. Hybrid processing has been used for applications such as information extraction and question answering. But to the best of our knowledge, the application of hybrid processing to grammar and style checking has not been previously investigated.

In this paper, we present an implemented prototype of a hybrid grammar and style checking system for German and English, called Checkpoint. As the baseline shallow system we have taken an industrial strength grammar and controlled language style checker, which is based on the FLAG technology. The deep processing platform used in the project is the PET parser (Callmeier, 2000) operating on wide-coverage English and German HPSG grammars, the English Resource Grammar (ERG) (Copestake and Flickinger, 2000) and the German Grammar (GG) (Müller and Kasper, 2000; Crysmann, 2005; Crysmann, 2007), respectively. The ERG and the GG have been developed for over 15 years and have already been used as deep processing engines in the Heart-of-Gold hybrid processing platform. We have developed an approach for the selective application of deep processing based on the error hypotheses of the shallow system. Error detection in the deep system follows a MAL-rule approach. In order to compare the benefits of the selective application of deep processing with its nonselective application, we have developed two scenarios: one parallel and one integrated. While the parallel (nonselective) scenario enables improvement in both recall and precision, the integrated (selective) scenario only enables improvement in precision. However, the performance of the integrated approach is much better. We have also investigated several possibilities of integrating deep processing in the selective scenario. Since the HPSG grammars are suitable both for parsing and generation, the system can successfully provide both error corrections and paraphrases of stylistic errors. For the purpose of investigation, evaluation and statistical parse ranking, we have collected and annotated several corpora of texts from technical manuals. Finally, the approach has been evaluated regarding error detection and performance.

2 The approach

Checkpoint has two main goals: (a) improving the precision and recall of existing pattern-based grammar and style checking systems for error types whose detection requires considering more than the strictly local syntactic context; and (b) generating error corrections for both grammar and style errors. Accordingly, we have chosen to focus on certain error types based on the difficulties of the pattern-based system.

2.1 Anticipation of grammar errors

Grammar errors are detected by means of error anticipation rules, or MAL-rules. MAL-rules exactly model errors, so that erroneous sentences can be parsed by the grammar. For this purpose we enlarged two HPSG grammars for German, the GG, and English, the ERG, with MAL-rules for error types that were problematic for the pattern-based shallow system. For German the following phenomena have been handled: subject verb agreement (*subject_verb_agreement*), NP internal agreement (*NP_internal_agreement*), confusion of the complementiser "dass" with the homophonous pronoun or determiner "das" (*dass_das*), as well as editing errors, such as local and non local repetition of words (*repetitions*). Here follow some examples (taken from the FLAG error corpus (Becker et al., 2000), and *die tageszeitung 'taz'*, a German newspaper):

(1) Auch in AOL gibt es Newsgroups, die dieses Thema *diskutiert* [=diskutieren]. (FLAG)
Also in AOL are there newgroups, which (Pl) this topic *discuss* (Sg).
'There are also newsgroups in AOL which discuss this topic.'

(2) Ich habe dem *ganze* [=ganzen] Geschehen von meinem Sofa aus zugesehen. (FLAG)
I have the *whole* (wrong adj. form) events from my couch out watched.
'I have watched the whole events from my couch.'

(3) Vor allem im Süden ... *führten* [=haben] die Liberalen der MR einen heftigen Wahlkampf gegen die PS *geführt*. (taz, June 2007)
Above all in the south ... *led* (past tense) the liberals of the MR a hard election campaign against the PS *led* (past participle).
'Particularly in the south, the liberals of the MR led a hard election campaign against the PS.'

For English, MAL-rules for errors concerning subject verb agreement and missing determiners were implemented.

2.2 Detection of stylistic errors

Stylistic errors are grammatical constructions that are dispreferred in a particular register or type of document. Sometimes certain constructions are not desirable because machine translation systems have problems dealing with them or because they prevent easy understanding. In such cases a controlled language approach is taken, where the problematic constructions are paraphrased into equivalent less problematic constructions. Since these constructions are grammatical they can be parsed and, thus, detected. A generation of a paraphrase is possible based on the semantic representation obtained through parsing. For German the following phenomena were handled: passive, future and implicit conditional sentences, as in the following example:

(4) Wartet man zulange, kriegt man keine Karten.
Waits one too long, gets one no tickets.
'If one waits too long one gets no tickets.'

Correct: Wenn man zulange wartet, kriegt man keine Karten.

For English we focussed on the following phenomena: passive (*avoid_passive*), future (*avoid_future*), modal verbs (*avoid_modal_verbs*), subjunctive (*avoid_subjunctive*), stand-alone deictic pronouns (*use_this_that_these_those_with_noun*) and clause order in conditional sentences (*condition_must_precede_action*).

2.3 Integrated vs. parallel scenarios

We have developed two integration scenarios: an integrated one and a parallel one. In the parallel scenario the pattern-based shallow system and the deep processing parser run independently of each other, that is, all sentences are parsed independent of whether the shallow system has found an error in them. In the integrated scenario the deep parser is only called for those sentences where the shallow system has detected some error of the type of those which Checkpoint is able to process (enumerated in subsection 2.1). The parallel scenario allows improvement in the recall of the shallow system, since Checkpoint can find errors that the shallow system has not found. In the integrated scenario, on the contrary, only the precision of the shallow system can be improved, since Checkpoint departs from the hypotheses of the shallow system. The integrated scenario, however, promises to perform better in time than the parallel scenario, since only a fraction of the whole text has to be scanned for errors. Moreover, the performance of the integrated system can also be improved with the selective activation of the MAL-rules that model the specific errors found by the shallow system. This greatly reduces the enormous search space of the parsing algorithms and the processing time resulting from the simultaneous processing of several MAL-rules.

The integration of the shallow system and the deep parser has been achieved through an extension of the PET parser that allows it to receive any kind of input information and integrate this into the chart. This preprocessing information can be, for example, part-of-speech tagging, morphology and lemmatisation, and already guides the parsing process. It allows, for instance, recognition of unknown words or identification of the correct lexical entry in cases where there is ambiguity. An input format in terms of feature structures, the "Feature Structure Chart" (FSC) format, has been developed for this purpose (Adolphs et al., 2008). The shallow system, thus, produces a feature structure chart, based on the information delivered by the various shallow modules, and this information is given as input to the PET deep parser, which reads it and integrates it into the chart.

Error hypotheses from the shallow system are passed to the deep parser by means of specific features in the input feature structure (MAL-features) of every input token in the FSC, permitting selective activation of MAL-rules. To this end, the original FSC generated by the shallow system, which contains information on the part-of-speech, the lemma and morphological features such as number, gender and case, will be extended with MAL-features. These MAL-features correspond to the class of some MAL-rule in the grammar and have boolean values. Signs in the grammar are specified for these MAL-features. MAL-rules are defined such that they can only take as their daughters edges with a positive value for the corresponding MAL-feature. All information in the FSC input tokens is passed to the tokens in the chart through a feature called TOKEN in lexical items. Thus, error hypotheses are passed from the input tokens to the lexical items in the chart by stating that the values of the MAL-features in the lexical items are

equal to the values of the MAL-features in the corresponding input tokens in the FSC.

The values of the MAL-features are obtained by checking the error report delivered by the shallow system. For certain errors detected by the shallow system there is a mapping to MAL-features. The value of a MAL-feature will be set to "+" if the shallow system has found the corresponding error. The rest of the MAL-features can be set to "bool" if we want to allow other MAL-rules to fire (which can improve recall, but increases ambiguity and, consequently, has a negative effect on performance). The values of the rest of the MAL-features can also be set to "-", if we want to prevent other MAL-rules from firing (which allows improvement only in precision, but limits ambiguity and, consequently, results in better performance). There is also the possibility of activating the relevant MAL-features only for those tokens which are, according to the shallow system, within the error span, instead of activating the MAL-features for all tokens in the erroneous sentence.

2.4 Generation of corrections and paraphrases

One of the advantages of using deep processing in grammar and style checking is the possibility of generating corrections and paraphrases which obey the constraints imposed by the syntactic context. Since the HPSG grammars that we are using are suitable both for parsing and generation, this is straightforward. Robust parsing delivers as output a semantic representation in the Minimal Recursion Semantics formalism (MRS) (Copestake et al., 2006) of the sentence which can be used for generation with the LKB (Carroll et al., 1999).

The MAL-rules directly assign well-formed semantic representations from which a correct surface string can be generated. In the case of stylistic errors, transfer rules are used to generate the desired paraphrase, using MRS-to-MRS mapping rules modelled on the semantic transfer-based machine translation approach of (Lœnning et al., 2004).

We identified two areas where generation of repairs will actually provide a considerable added value to a grammar checking system: first, for non-native speakers, simple highlighting of the error location is often insufficient, since the user may not be familiar with the rules of the language. Second, some areas, in particular stylistic ones may involve considerable rearrangement of the entire sentence. In these cases, generation of repairs and paraphrases can reduce editing cost and also minimise the issue of editing errors associated with non-local phenomena.

The generator and HPSG grammars we use are able to provide a range of realisations for a given semantic input. As a result, realisation ranking is of utmost importance. In order to select repairs which are both smooth and maximally faithful to the input, modulo the error site, of course, we combined two methods: a discriminative PCFG-model trained on a generation treebank, enhanced by an n-gram language model, cf. (Velldal and Oepen, 2005), and an alignment approach that chooses the most conservative edit from a set of input realisations. As our similarity measure, we employed a variant of BLEU score (NEVA), suggested in (Forsbom, 2003). The probabilistic ranking models we trained achieve an exact match accuracy of 73% for both English (Velldal and Oepen, 2005) and German (as evaluated on the subset of TiGer the error corpus was based on).

3 Error corpora

In order to learn more about the frequencies of the different error types, to induce statistical models that allow us to obtain the best parse in the domain of technical manuals and to evaluate our implemented approach to grammar and style checking, we collected and manually annotated corpora from the domain of technical documentation.

Since errors in pre-edited text tend to be very scarcely distributed, manual annotation is quite costly. As a result, instance of certain well-known error types cannot be tested in a greater variety of linguistic environments. To overcome this problem, we semi-automatically derived an additional error corpus from a treebank of German.

English For purposes of evaluation in a real world scenario, we constructed a corpus for English, consisting of 12241 sentences (169459 words) from technical manuals. The corpus was semi-automatically annotated with several types of grammar and style errors. For this purpose annotation guidelines were developed, which contained the description of the errors together with examples of each and their possible corrections. The annotation took place in two phases. First, we wanted to find out about the precision of the shallow system, so we ran the shallow system over the data. This resulted in an annotation for each error found consisting of the erroneous sentence, the error span and the type of error. The annotators, who were native speakers, then decided whether the errors had been correctly detected. In the second phase, we aimed to create a gold standard, so as to be able to evaluate both the shallow system and Checkpoint regarding recall and precision. For this purpose, we extracted the errors that had been annotated as correctly detected in the previous phase and the annotators only had to find the non-detected errors in the rest of the corpus. For the latter, they also marked the span and identified the error type.

Subsets of these two datasets were treebanked with the corresponding HPSG grammars. We employed the treebanking methodology developed for Redwoods (Oepen et al., 2002), which involved

first parsing a corpus and recording for each item the alternative analyses (the parse forest) assigned by the grammar, then manually identifying the correct analysis (if available) within that parse forest. This approach provides both a gold standard syntactic/semantic analysis for each parsed item, and positive and negative training data for building an accurate statistical model for automatic parse selection.

German For German, we pursued a complementary approach towards corpus construction. Here the focus lay on creating a test and evaluation corpus that provided instances of common error types in a variety of linguistic contexts. Since manual error annotation is highly costly, owing to scarce error distributions in pre-edited text, we chose to automatically derive an error corpus from an existing treebank resource. As for the error types, we focussed on those errors which are arguably performance errors, as e.g. missing final consonants in inflectional endings, the confusion of homophonous complementiser and relative pronoun, or else, editing errors, such as local and non-local duplicates.

We introduced instances of errors in a subcorpus of the German TiGer treebank (Brants et al., 2002), nicknamed TiG-ERR, consisting of 77275 words (5652 sentences) from newspaper texts. All the sentences in this subcorpus were parsable, so that an evaluation of Checkpoint in the ideal situation of 100% coverage could be carried out. The artificially introduced errors were of the following types: *subject_verb_agreement*, *NP_internal_agreement*, *dass/das*, and *repetitions*, all of them already illustrated with examples in section 2.1.

Additionally, we annotated a corpus of technical documents for these error types to estimate the distribution of these error types in pre-edited text.

4 Error models

In order to construct a statistical parse-ranking model which could determine the intended use of a MAL-rule in the analysis of a sentence where the grammar produced analyses both with and without MAL-rules, the English treebank was constructed using the version of the ERG which included the MAL-rules. 4000 sentences from the English corpus were presented to the parser, of which 86.8% could be parsed with the ERG, and of these, the annotators found an intended analysis for 2500 sentences, including some which correctly used MAL-rules. From these annotations, a customised parse selection model was computed and then used in parsing all of the corpus, this time recording only the one analysis determined to be most likely according to this model. We also compared accuracy of error detection based on this new model with the accuracy of a pre-existing parse-selection model trained on tourism data for LOGON, and confirmed that the new model indeed improved over the old one.

For German, we have not created a specific statistical model yet, but, instead, we have used an existing parse selection model (Crysmann, 2008) and combined it with some heuristics which enable us to select the best error hypothesis. The heuristics check for each parsed sentence whether there is an analysis containing no MAL-rule. If there is one and this is not ranked as the best parse, it is moved to the first position in the parse list. As a result, we can eliminate a high percentage of false alarms.

5 Evaluation results

We have evaluated the English and the German versions of Checkpoint against the corpora described in section 3.

German For German we have taken as a test corpus standard the TiG-ERR subcorpus containing the automatically introduced errors, and have parsed all its sentences. The following table shows the frequencies of the different types of handled errors in the corpus of technical manuals, the FLAG error corpus (Becker et al., 2000), and in the TiG-ERR corpus. The electronic version of the FLAG corpus consists of 14,492 sentences, containing 1,547 grammar or style errors.

ERROR TYPE	MANUALS	FLAG	TiG-ERR
NP_internal_agr	119	180	2258
subject_verb_agr	17	63	748
dass/das	1	152	75
repetitions	19	n/a	2571

Table 1: Frequencies of the error types for German

The following charts show the values for recall and precision for the shallow system and Checkpoint. As you can see, Checkpoint improves the recall for the error types *subject_verb_agreement* and *NP_internal_agreement*, whereas the precision remains more or less the same. For the error type *dass/das* Checkpoint improves both recall and precision. For the error type *repetitions*, which is only partially handled by the spell checker in the shallow system, Checkpoint reaches considerable recall and precision values.

Deep processing on average improves the recall of the shallow system by 21% and the precision remains equal at 0.83. According to the error frequencies in the corpus of technical manuals, deep processing would improve the recall of the shallow system by only 1.7%, since the error types *subject_verb_agreement*, *NP_internal_agreement* and *dass/das* only make up 6.57% of the total amount of annotated errors. However, as we found out later, the corpora of technical manuals consist of texts that have already undergone correction, so the errors are very sparse.

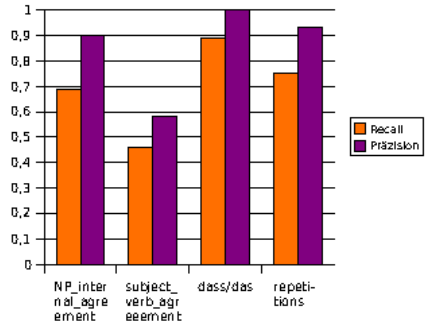

Figure 1: Checkpoint values for recall and precision for German

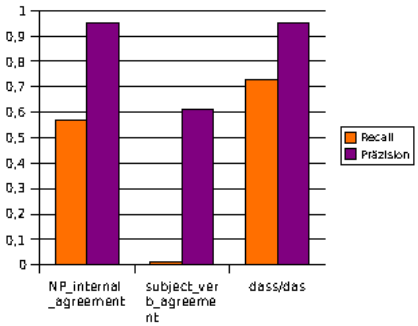

Figure 2: Values for recall and precision for the shallow system for German

Through the MAL-rules the coverage of the GG on the TiG-ERR corpus increased to 85% - 95%, whereas without the MAL-rules the coverage was 10%. This 10% coverage included overgeneration by the grammar, as well as sentences that, after the automatic insertion of errors, still remained grammatical, although they didn't express the intended meaning any more.

The performance of the parallel and integrated scenarios was compared. The ambiguity of the MAL-rules, that is, the possibility of applying several MAL-rules to a unique error, considerably deteriorates the performance when processing sentences containing several errors. In a subcorpus containing *NP_internal_agreement* errors, the average processing time per sentence increases from 8.3 seconds with the selective activation of MAL-rules to 31.4 seconds with the activation of all MAL-rules. Particularly the MAL-rules modeling the error *subject_verb_agreement* are a source of ambiguity. If these MAL-rules are only selectively activated the average processing time per sentence decreases to 14.9 seconds.

Finally, we have evaluated the performance of the German grammar in the task of error correction, using non-local duplicates and adjectival agreement errors as a test bed. For these error types, the German HPSG grammar generated repairs for 85.4% of the detected non-local duplicates and 90% of the detected agreement errors.

English For English we have only implemented and evaluated the parallel scenario. The focus for English evaluation was the recognition of those stylistic errors whose correction requires a restructuring of the sentence, and the generation of the corresponding paraphrases. The recognition of such error types is not based on MAL-rules, but on certain already existing rules in the grammar. The approach was evaluated taking the manually annotated English corpus of technical manuals as a gold standard. The following table shows the frequencies of the error types handled by Checkpoint.

ERROR TYPE	OCCURRENCES
avoid_future	404
avoid_modal_verbs	657
avoid_passive	213

Table 2: Frequencies of the error types for English

The PET parser with the ERG reached 86.1% coverage on the full corpus. The following charts show the values for recall and precision for Checkpoint and the shallow system.

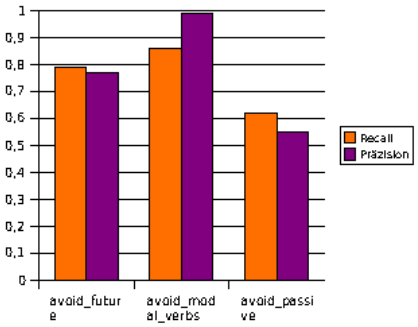

Figure 3: Checkpoint values for recall and precision for English

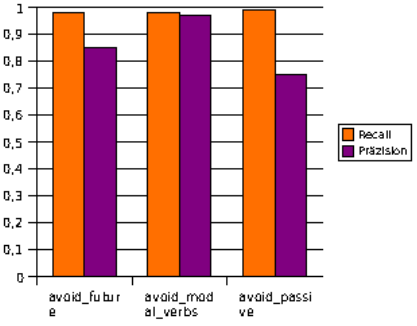

Figure 4: Values for recall and precision for the shallow system for English

As one can see, for the stylistic errors *avoid_future* and *avoid_modal_verbs*, Checkpoint reaches values which, although relatively high, are lower than the shallow system. In most cases a paraphrase for these errors can be constructed, so the improvement Checkpoint provides here is the generation of corrections. For the error type *avoid_passive* the precision is not so high, which is due in part to mistakes in the manual annotation. The passive sentences found by Checkpoint are actually passive sentences. However, these were

not annotated as passives, because the annotators were told to annotate only those stylistic errors for which a paraphrase was possible. The same happens for stylistic errors like *avoid_subjunctive*, *use_this_that_these_those_with_noun* and *condition_must_precede_action*. In principle, Checkpoint is very good at finding these types of errors, but we cannot yet present a reliable evaluation here, since only those errors were annotated for which a paraphrase was possible. This approach is reasonable, since no error alarm should be produced when there is no other possibility of expressing the same. However, since we have not yet developed a method which allows us to automatically distinguish those cases for which a paraphrase is possible from those for which none is, we would need to annotate all occurrences of a phenomenon in the corpus, and introduce a further annotation tag for the paraphrase potential of the sentence.

Nevertheless, even if the grammar-based research prototype cannot beat the industrial pattern-based system in terms of f-measures, we still believe that the results are highly valuable in the context of our integrated hybrid scenario: Since the full reversibility of the ERG has already been established independently by (Velldal and Oepen, 2005), the combined system is able to generate error correction for a great proportion of the errors detected by the shallow component. This includes 80% and above for *avoid_future* and *avoid_modal_verbs*.

6 Summary and conclusions

In this paper we have presented an implemented approach to grammar and style checking based on hybrid processing. The hybrid system has two components: a shallow grammar and style checking system based on the FLAG technology, and the PET deep parser operating on linguistically motivated grammars for German and English. The German version of the hybrid system improves the recall and in certain cases the precision of the shallow system and generates error corrections. For English, the hybrid system in most cases successfully generates paraphrases of sentences containing stylistic errors. Although we only have explored some of the possibilities of integrating deep and shallow processing for the grammar and style checking application, these results speak for the feasibility of using hybrid processing in this task.

We have developed an integrated strategy which forwards the output of the shallow system, including both the output from several pre-processing linguistic modules and the error hypotheses, as input to the deep parser. This procedure not only improves the robustness of the deep parser with the recognition of unknown words and reduces ambiguity by instantiating only those lexical items consistent with the hypotheses of the POS tagger or the morphology; but it also allows the selective application of grammar rules, which considerably reduces the search space for parsing and, consequently, improves performance. Based on the error hypotheses of the shallow system, the selective application of grammar rules is achieved by positing features in the Feature Structure Chart whose particular values are a pre-condition for MAL-rules to apply. The improvement in performance suggests that this strategy can be extensible to parsing in general based on pre-processing components. Given the output of a chunker, for example, certain syntactic configurations can already be excluded. Having features whose values allow one to switch off certain rules not compatible with these configurations would considerably reduce the search space.

On the other hand, we have run the two modules independently from each other to find out how the recall of the shallow system can be improved by deep processing. The fact that for several error types, such as *subject_verb_agreement* and *NP_internal_agreement*, recall can be considerably improved suggests that, in order not to parse all sentences, the shallow system should send an error hypothesis to the deep system when finding particular syntactic configurations which may indicate the occurrence of such errors. In this way, such error hypotheses, although not reliably detectable by the shallow system alone, could be confirmed or discarded with a focussed application of deep processing, which would not be as resource consuming as parsing every sentence.

One of the results of the experiment has been an on-line demonstration system. The running system shows that the different modules can be easily combined with each other. Our hybrid approach, however, is generic and portable. Although implemented for our specific baseline system, it can in principle be used with other shallow systems.

Acknowledgements

The research reported in this paper has been carried out as part of the DFKI project Checkpoint, running from February until November 2007. The project was funded by the ProFIT program of the Federal State of Berlin and the EFRE program of the European Union.

References

Adolphs P., S. Oepen, U. Callmeier, B. Crysmann, and B. Kiefer. 2008. Some Fine Points of Hybrid Natural Language Parsing. Proceedings LREC-2008, Marrakech, Morocco.

Becker M., A. Bredenkamp, B. Crysmann, and J. Klein. 2003. Annotation of error types for a German newsgroup corpus. In A. Abeillé, editor, *Treebanks. Building and Using Parsed Corpora*, number 20 in Text, Speech And Language Technology. Kluwer, Dordrecht.

Bender, E. M., D. Flickinger, S. Oepen, A. Walsh, and T. Baldwin. 2004. Arboretum: Using a precision grammar for grammar checking in call. In *InSTIL/ICALL symposium 2004. NLP and speech technologies in advanced language learning systems.* Venice, Italy.

Brants, T., S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol.

Bredenkamp, A., B. Crysmann and M. Petrea. 2000. Looking for errors: A declarative formalism for resource-adaptive language checking. In *Proceedings LREC-2000*. Athens, Greece.

Bredenkamp, A., B. Crysmann and M. Petrea. 2000. Building multilingual controlled language performance checkers. In *Proceedings of the CLAW 2000*. Seattle, WA.

Callmeier, U., A. Eisele, U. Schäfer, and M. Siegel. 2004. The Deepthought core architecture framework. In *Proceedings of LREC-2004*, 1205–1208, Lisbon, Portugal.

Callmeier, Ulrich. 2000. PET — a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering* 6(1):99–108.

Carroll, John and Ann Copestake and Dan Flickinger and Victor Poznanski. 1999. An efficient chart generator for semi-lexicalist grammars. *Proceedings of ENLG*, pp. 86–95.

Copestake, A., and D. Flickinger. 2000. An open-source grammar development environment and broad-coverage english grammar using HPSG. In *Proceedings LREC-2000*. Athens, Greece.

Copestake, A., D. Flickinger, C. Pollard, and I. Sag. 2006. Minimal recursion semantics: an introduction. *Research on Language and Computation* 3(4):281–332.

Crysmann, B., A. Frank, B. Kiefer, S. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, F. Xu, M. Becker, and H.-U. Krieger. An integrated architecture for shallow and deep processing. In *Proceedings of ACL 2002*, University of Pennsylvania, Philadelphia, 2002.

Crysmann B. 2005. Relative clause extraposition in German: An efficient and portable implementation. *Research on Language and Computation*, 3(1):61–82.

Crysmann B. 2007 Local ambiguity packing and discontinuity in German. In T. Baldwin, M. Dras, J. Hockenmaier, T. H. King, and G. van Noord, editors, *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, pages 144–151, Prague, Czech Republic.

Crysmann B. 2008. Parse Selection with a German HPSG Grammar. In S. Kübler and G. Penn, editors, *Proceedings of the ACL 2008 Workshop on Parsing German (PaGe)*, pages 9–15, Columbus, Ohio, USA.

Douglas, S. 1995. Robust PATR for error detection and correction. In A. Schoeter and C. Vogel (Eds.)*Nonclassical feature systems*, Vol. 10, pp. 139-156. Centre for Cognitive Science, University of Edinburgh.

Forsbom, E. 2003. Training a Super Model Look-Alike. *Proceedings of the MT Summit IX Workshop "Towards Systemizing MT Evaluation", pp. 29-36*.

Gaizauskas, R., H. Cunningham, Y. Wilks, P. Rodgers and K. Humphreys. 1996. GATE: An environment to support research and development in natural language engineering. In *Proceedings of the 8th IEEE international conference on tools with artificial intelligence.* Toulouse, France.

Jensen, K., G. E., Heidorn and S. D. Richardson (Eds.). 1993. Natural language processing: The PLNLP approach. Boston - Dordrecht - London.

Haller, J. 1996. MULTILINT: A technical documentation system with multilingual intelligence. In *Translating and the computer 18*. London.

Heinecke, J., J. Kunze, W. Menzel, and I. Schroeder. 1998. Eliminative parsing with graded constraints. In *Proceedings ACL/Coling 1998*, Vol. I, pp. 526-530. Universite de Montreal, Montreal, Quebec, Canada.

Lœnning J. T. , S. Oepen, D. Beermann, L. Hellan, J. Carroll, H. Dyvik, D. Flickinger, J. B. Johannessen, P. Meurer, T. Nordgård, V. Rosén and E. Velldal. 2004. LOGON. A Norwegian MT effort. In *Proceedings of the Workshop in Recent Advances in Scandinavian Machine Translation.* Uppsala, Sweden.

McCoy, K. F., C. A. Pennington, and L. Z. Suri. 1996. English error correction: A syntactic user model based on principled "mal-rule" scoring. In *Proceedings of UM-96, the Fifth International Conference on User Modeling*, pp. 59-66. Kona, Hawaii.

Menzel, W. 1998. Constraint satisfaction for robust parsing of natural language. In *Theoretical and Experimental Artificial Intelligence*, 10 (1), 77-89.

Müller, S., and W. Kasper. 2000. HPSG analysis of German. In W. Walster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, 238–253 Springer, Berlin.

Oepen, S., K. Toutanova, S. Shieber, C. Manning, D. Flickinger and T Brants. 2002. The LinGO Redwoods Treebank. Motivation and Preliminary Applications. In *Proceedings of COLING 2002*. Taipei, Taiwan.

Ravin, Y. 1998. Grammar errors and style weaknesses in a text-critiquing system. In *IEEE Transactions on Communication*, 31 (3)

Schmidt-Wigger, A. 1998. Grammar and style checking for German. In *Proceedings of CLAW 98*. Pittsburgh, PA.

Velldal, E. and S. Oepen. 2005. Maximum entropy models for realization ranking. In *Proceedings of the 10th MT-Summit (X), Phuket, Thailand*.