Chapter 1

ANNOTATION OF ERROR TYPES FOR GERMAN NEWS CORPUS

Markus Becker, Andrew Bredenkamp, Berthold Crysmann, and Judith Klein
DFKI GmbH, Saarbrücken

1. INTRODUCTION

This paper will discuss the corpus annotation effort in the FLAG project and its application for assisting in the development of controlled language and grammar checking applications.

The main aim of the German government funded FLAG project[1] is to develop technologies for controlled language (CL) and grammar checking applications for German. The project work has therefore been divided into two separate but complementary streams of activity. Firstly, the aim was to develop an modular NLP software architecture for quickly developing different kinds of CL and grammar checking applications. Secondly, to validate the first activity, it was seen as important to build up an empirical base for testing and formally evaluating checking components. Given the lack of existing annotated corpora of errors for German (or indeed for any language as far as the authors know), the construction of such a corpus was a high priority task. This would enable us not only to perform quantitative tests, but also to derive an empirically based typology of errors which the project could use for orientation.

The corpus was particularly important given the approach which the FLAG project was taking to the task of grammar and controlled language checking, which relies on a phenomenon-oriented approach to the problem of identifying errors, using shallow processing techniques. In order to fine-tune the heuristics which are central to such an approach, i.e. one based on identifying "candidate errors" of increasing probability, it is essential to have good test suites annotated with respect to the phenomena under investigation.

The annotation of the corpus was to be carried out in such a way that we could easily access and quantify snapshots of the data, for producing test suites for testing purposes and for producing statistics on the frequency of particular error types.

The research community not only lacked an annotated corpus of errors, there was no existing ontology of errors which could be easily translated into an annotation schema. The definition of such a schema based on traditional descriptions of errors (such as Luik, 1993a; Luik, 1993b) thus formed the first major workpackage. Fortunately, tools for the annotation of corpora, and the management thereof are becoming increasingly sophisticated; it was therefore necessary to evaluate a number of tools in the light of our specific needs.

2. CORPUS DESCRIPTION

Given the use we intended to make of the language data, the corpus had to be carefully selected. The availability of unedited corpora with real errors proved to be somewhat problematic, since most available sources of texts, newspapers being a traditional source for such things, are rather heavily edited before they are made available. Our corpus needed the following characteristics:

- the texts should have a relatively high error density, i.e. as close to a real performance scenario as possible. It was important to avoid annotators having to annotate (or rather identify) too many grammatical sentences in building up a useable resource of ungrammatical ones.
- they should be easily accessible - the texts needed to be easily available in large quantities such that we could be assured of having a large quantity of errors for all error types.
- they should be electronically available - to facilitate the electronic manipulation (e.g. tokenization) and annotation of the corpus using the software tools, and for storing the results in a database to facilitate the querying of the data, for the production of test suites, etc.

The corpus selected was an archive of email messages posted to a number of internet USENET newsgroups. The initial selected corpus contained approximately 120,000 sentences, although additional text is added constantly to the archive and can be retrieved automatically as required. The corpus consists of email messages of various lengths and with varying degrees of register, although the tendency is clearly towards a rather colloquial style. The main benefit of the corpus is that it is largely unedited text with a relatively high proportion of grammatical errors, both performance errors and competence errors.

3. ANNOTATION STRATEGY

The annotation proceeded in three main stages: the first stage was to develop a typology for grammatical errors for German. Subsequently, we undertook manual annotation on paper with a simplified error typology, the idea being that

we could get a large set of sentences annotated quickly. This proceeded very rapidly and allowed us to focus on the annotation of ungrammatical sentences in the third stage, the annotation using computer tools.

3.1 ERROR TYPOLOGY

The error typology took as its basis traditional grammar books and an collection of cases considered grammatically difficult, cf. Stolpersteine (Luik, 1993a; Luik, 1993b). We produced a fine-grained error typology with various sub-classes for orthographic, morphological, morpho-syntactic, syntactic and syntactic-semantic errors (Crysmann, 1997). This typology was taken as the starting point for the annotation, with the intention that it would evolve in the light of the experience gained from the detailed inspection of the corpus.

In order to support the integration of new error types derived in the course of the annotation effort, the initial error typology has been designed as a type hierarchy, including some highly general supertypes in addition to the initial set of more specific error types.

In addition to the classification of error types, the typology also presents some preliminary ideas on error domains, in that it tries to define the relation between the affected words. This information could be of use in guiding checkers to detect errors, and could also help find the right correction/proposal on the basis of a diagnosis.

Finally, the typology also makes the notion of "lexical anchors", i.e. words which themselves are often the focus of a particular error type (cf. *Wissens* 'knowledge' in the contrast **meines Wissens nach* 'my.GEN knowledge.GEN according' vs. *meines Wissens* 'as far as I know; lit.: my.GEN knowledge.GEN' or *meinem Wissen nach* 'as far as I know; lit.: my.DAT knowledge.DAT according').

Given the time-consuming task of annotation a more coarse-grained version with 16 error types was used for the initial annotation task, as follows:

- one class for morphological errors (e.g. derivation errors),
- syntax errors comprise word order, categorial, case, and three classes for agreement errors,
- one class for syntactic-semantic selection (as for fixed verbal structures),
- four types of orthographic errors,
- one further general class for syntax errors which can't be easily classified further.

(Subsequent classes which are less interesting for our immediate purposes are: a class for obvious tokenizer mistakes, one for typographical errors, e.g.

repeated words, and one for competence errors which can't easily be reconstructed (corrected) to form grammatical sentences.)

3.2 MANUAL ANNOTATION

When we started the annotation task in FLAG there were no customised annotation tools available. The sentences from the raw corpus were tokenized and presented in a format which facilitated the process of annotation, each sentence being printed with a number of boxes corresponding to the broad error categories we were using at this stage. The annotators had simply to identify the errors, mark the error position and check the appropriate error label box.

During this annotation phase, specific and very complex errors (and also ambiguities in analysing/correcting the errors) occurred within the emails sentences. Guidelines were established and the typology was further refined to help with consistent annotation.

3.3 COMPUTER ANNOTATION

To best exploit the annotated news corpus, the email sentences needed to be electronically available and easily accessible. In the second phase of the annotation work the manually annotated sentences are being processed with the help of the two annotation tools. This computer annotation is being done by another annotator in order to validate the annotations and apply the latest version of the elaborated annotation guidelines.

This work serves also to help validate and customise the annotation tools themselves. One the basis of our experiences using two different tools, it was decided to focus the computer annotation effort on a single system, namely the DiET tool. The other tool evaluated, Annotate, whilst offering much more sophisticated annotation functionality for syntactic trees, was less well suited to dealing with the flexible ontology with which we were working.

As soon as all manually annotated email sentences are validated and recorded, new corpus data will directly be annotated with the DiET annotation tool. Even though the identification of errors on screen is slightly more error-prone than working on paper, the direct methods is more economical.

Additionally, the search facility of DiET will be used to systematically check all annotations.

So far about 60.000 sentences have been annotated on paper, of which over 14,492 have now been recorded and checked in the DiET tool. Out of these, 6473 contain at least one error. More than 3900 sentences were annotated using the Annotate tool, which will be imported automatically into DiET.

4. ANNOTATION TOOLS

As mentioned above, two sophisticated annotation tools were evaluated: Annotate, the corpus annotation tool of the Negra project (`http://www.coli.uni-sb.de/cl/projects/negra.html`) and DiET, the multi-purpose annotation tool developed within the DiET project (`http://diet.dfki.de/`). The feedback we were able to provide has lead to a number of modifications and extensions in the tools.

Annotate is a Unix-based annotation tool which has been explicitly designed for the morpho-syntactic annotation of text corpora. It is equipped with a graphical user interface for efficient tree annotations and uses a relational database for storage and retrieval. In order to use the tree format for the error annotation Annotate has been customised such that the tree nodes provide the labels for the error types and the tree edges provide descriptive information on particular error types. The error position is indicated by the edges and nodes build upon the words. With Annotate we did first experiments for illustrating the error domains. The tree structure has been used in order to build the dependency relation between the words which are affected by the error.

While the tree format employed in Annotate provides for a rich representation of the structure of a particular error in terms of relations (cf. Figure 1.1), its major drawback lies in the fact that these representations have to be built up from bottom to top, with one of the most basic pieces of information, i.e. the error type, added last. Similarly, all dimensions of the annotation scheme, including locality, type, and domain, have to be entered in a single step, because integration of additional annotations at a later point cannot be performed without reconstructing major subparts of the annotation tree.

Furthermore, the tree-structural organisation does not offer a convenient solution to the representation of overlapping errors.

DiET is a comprehensive software package for the construction, annotation, customisation, and maintenance of structured reference data. The system is a Java application, implemented in a configurable, open client/server architecture with a central database system managing the data and a client integrating construction, annotation, and retrieval facilities. DiET allows the user to easily configure an annotation schema by specifying relevant annotation types (i.e. describing attributes for annotation) with adequate data types (numbers, strings, marking, trees, etc.). The marking mode is especially relevant for annotating the error locality within the sentences.

As already mentioned, we decided to continue the computer annotation only on DiET which has substantially improved in the meantime. It now offers also a sophisticated tree grapher for illustrating the error dependencies within the error domains.

Figure 1.1 Error annotation in ANNOTATE

The FLAG annotation schema for error types contains the following information:

- email number and sentence number to identify the origin of the data,
- sentence status (grammatical, ungrammatical, colloquial),

Figure 1.2 Error annotation in DIET

- error type (orthography, subject-verb agreement, word order, etc.) ,
- number of errors per sentence,
- error locality (at what sentence position the error occurs),

- error domain (dependency relation between the affected words).

The schema mainly reflects the manual annotation format and thus allows the annotators to easily transfer the manually annotated emails into the DiET database.

Although the tree grapher integrated in DiET provides a representation slightly less expressive than the one in Annotate, most (if not all) aspects of the error annotation schema can be mimicked using DiET's concept of multiple annotation views. These views not only provide an elegant solution to the problem of overlap, they also support splitting up the entire annotation into several logical subtasks, e.g. localisation and classification as basic annotations, which are then complemented by error domains at some later stage.

The retrieval facilities of DiET will be used in order to systematically check the annotations. With specific database queries it can be checked if there are entries where the sentence status is grammatical but there is also an entry for error type or it can be search for sentences which have an entry for error type but no entry for number of errors per sentence.

5. EVALUATION

The original motivation for our annotation effort derived mainly from the need to verify the phenomenon-based error typology developed so far in the FLAG project. To process as much material as possible, annotators were given distinct subsets of the corpus for proof-reading and classification. However, when the results of the paper annotation were transferred into the DiET database, the classification was checked by a different annotator, thereby ensuring a certain level of precision and homogenity among the annotations.

In order to evaluate the usability of the corpus as a test suite, we conducted an experiment to measure the interjudge agreement, a valuable measure in the context of annotation tasks, on the basis of which recall and precision can be determined.[2] To achieve this, we handed out instances of the same subset of the corpus to our two main annotators. To get a realistic result, the annotators were given the directive to have no clarification dialogues during annotation. After the annotation, the two suite instances were frozen. A third instance of the same suite will be created as the result of a discussion of differently annotated items between the two annotators. Currently, we do not have this reference corpus yet, so we have to confine our discussion to the direct comparison of the two independently annotated corpora.

The annotation task consists of two different subtasks, namely error detection and error classification. Thus, in the absence of a reference corpus, we can still determine a preliminary measure of each annotator's recall, by comparing the error detection rates: to achieve this, we identified the total number of detectable errors as the set union of the errors found by the two annotators. Table 1.1

summarises the results that were determined on a sentence by sentence basis, ignoring minor differences as to the exact position at which the error was identified.

		Annotator A		*Annotator B*	
Label	*Total*	*Abs.*	*Rel.*	*Abs.*	*Rel.*
S	0	0	–	0	–
SASV	3	2	0.67	3	1.00
SAAA	0	0	–	0	–
SC	16	12	0.75	12	0.75
SO	6	3	0.50	4	0.67
SG	0	0	–	0	–
SGCas	10	7	0.70	6	0.60
SGCat	71	51	0.72	64	0.90
SGS	34	25	0.74	24	0.71
Σ Syntax	140	100	0.71	113	0.81
O	164	130	0.79	145	0.88
OI	193	173	0.90	169	0.88
OS	116	85	0.73	84	0.72
OC	184	170	0.92	153	0.83
M	17	8	0.47	15	0.88
Σ Orthography	647	566	0.87	566	0.87
P	16	12	0.75	12	0.75
T	60	49	0.82	51	0.85
All	890	727	0.82	744	0.84

Table 1.1 Error detection (Recall)

The original subset of the corpus contained 745 sentences of which both annotators processed about 690 sentences. The remaining approximately 50 sentences were judged completely inacceptable and deleted, e.g. due to tokenization errors. Annotator A marked 727 errors, while 744 errors were found by annotator B. While the overall recall is above 80 % for both annotators, annotator A has a slightly better recall with orthographical errors than with grammar errors, whereas no significant difference between these two major classes can be found for annotator B. Given that there is often no clearly defined boundary between stylistic and grammatical errors, it is hardly surprising

that the class of syntax errors shows a higher degree of divergence than the class of orthographical errors.

In order to measure the degree of interjudge agreement concerning error classification, we compared the set of 478 errors which were marked at exactly identical positions within the 690 sentences. Of these, 454 errors were assigned the same error type. In other words, whenever two annotators find the same error they have a widely agreeing intuition about the appropriate error type, i.e. we have a high precision rate of around 95%. The error types assigned to the remaining set of 24 errors were distributed quite equally across the different error categories, including orthographical and grammatical errors. We interpret the high level of interjudge agreement regarding the error classification as a confirmation of the error typology developed in FLAG.

6. FIRST RESULTS

So far, a subset of the corpus containing 14,492 sentences has been fully annotated using the DiET tool. Among these, 7392 sentences were well-formed, with another 627 sentences being considered badly tokenised. The remaining 6473 sentences thus contained at least one error. Table 1.2 gives a summary of the distribution of errors found in these sentences.

The first major result that can be derived from the inspection of Table 1.2 concerns the relative scarcity of true grammar errors. Even though the overall density of errors is comparatively high, mainly a result of the choice of corpus, affecting almost 50 % of all sentences, the vast majority (around 83%) are purely orthographical errors. Grammar errors, by contrast figure around 16%, only.

If we have a closer look at the distribution within the class of syntax errors itself, we find that subcategorisation errors make up the bulk of the grammatical errors found in the corpus (around 9.4%). Roughly two thirds of these errors are target-deviant elisions (6.1%). Another major subclass of subcategorisation errors involves the erroneous use of the complementiser *daß* 'that' and the homophonous relative pronoun *das* 'which': in 49 cases the complementiser was used in place of a relative pronoun, while 103 times the relative pronoun was found instead of the complementiser, yielding a frequency of 1.7% for this specific error.

In contrast to subcategorisation errors, all other syntax errors only display a frequency of 3% and below. Among these, there are classes which are practically unattested, e.g. antecedent-anaphor agreement (SAAA) with only a single occurence in the entire set of 14,492 sentences. Results such as this are particularly useful in the design of error-checking applications, as they enable us to ignore very low frequency error types in case they would otherwise demand an inappropriately high level of processing.

Error type	*Label*	*Token*
Syntax (general)	S	3
Subject-verb agreement	SASV	63
Antecedent-anaphor agreement	SAAA	1
Concord (NP-internal agreement)	SC	180
Word order	SO	79
Valency (general)	SG	0
Subcategorisation	SGCat	854
Case assignment	SGCas	102
Semantic selection	SGS	265
Σ Syntax		1547
Morphology	M	91
Othography (general)	O	2893
Punctuation	OI	1701
Capital vs. small letters	OC	2776
One word vs. separate words	OS	1100
Σ Orthography		7561
All		9108

Table 1.2 Distribution of Error Types

The data extracted from the annotated test-suite can be useful to guide not only the detection of errors, but also the direction in which errors should be corrected: subject-verb agreement, for example, involves the verb as the error site in 56 out of 63 cases.

The relative scarcity of grammar errors in general and their uneven distribution across the different subtypes of syntax errors appear to support the focussed, phenomenon-based approach to error-checking adopted within the FLAG project, where inexpensive shallow processing technology is used to systematically identify error candidates, which are then subjected to more elaborate processing.

A final observation concerns the fact that annotators did not make much use of the more coarse-grained supertypes, such as S or SG. Along with the high precision rate in interjudge agreement discussed in the previous section, this fact appears to suggest that the error types identified in the FLAG annotation scheme are already sufficient to cover the relevant phenomena.

7. CONCLUSION

The corpus annotation effort in the FLAG project is a central part of the research effort, without the resources it provides it is not practically possible to optimise the software components under development. The methodology (fast paper annotation, followed by thorough evaluation in the course of computer annotation) is proving effective, and the "genuine" (i.e. corpus-derived) test-suites makes for more realistic, but nevertheless focussed testing.

Notes

1. The project is funded by the German Ministry for Education and Research (BMBF) Project number ITW 9700.

2. Of course, the interjudge agreement so determined, will also provide us with the upper bound in the context of evaluating the performance of automatic error checking components.

References

Crysmann, B. (1997). Fehlerannotation. Technical report, DFKI GmbH.

Klein, J., Lehmann, S., Netter, K., and Wegst, T. (1998a). Construction and annotation of test-items in DiET. In *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation*, Saarbrücken.

Klein, J., Lehmann, S., Netter, K., and Wegst, T. (1998b). DiET in the context of mt evaluation. In Nübel, R. and Seewald-Heeg, U., editors, *Evaluation of the Linguistic Performance of Machine Translation System*, pages 107–126. ???, St. Augustin.

Luik, G. (1993a). *Stolpersteine*, volume 1. ???, Wiesbaden.

Luik, G. (1993b). *Stolpersteine*, volume 2. ???, Wiesbaden.

Netter, K., Armstrong, S., Kiss, T., Klein, J., Lehmann, S., Milward, D., Regnier-Prost, S., Schäler, R., and Wegst, T. (1998). DiET — diagnostic and evaluation tools for natural language processing applications. In *Proceedings of the First International Conference on Language Resources and Evaluation*, pages 573–579, Granada.

Skut, W., Krenn, B., Brants, T., and Uszkoreit, H. (1997). An annotation scheme for free word order languages. In *Proceedings of ANLP*, pages 8–96, Washington.