Grammar checking for Swedish second language learners

Johnny Bigert Viggo Kann Ola Knutsson Jonas Sjöbergh[*]
KTH Nada, SE-100 44 Stockholm

Abstract

Grammar errors and context-sensitive spelling errors in texts written by second language learners are hard to detect automatically. We have used three different approaches for grammar checking: manually constructed error detection rules, statistical differences between correct and incorrect texts, and machine learning of specific error types.

The three approaches have been evaluated using a corpus of second language learner Swedish. We found that the three methods detect different errors and therefore complement each other.

Svensk sammanfattning

Grammatikfel och kontextberoende stavfel (felstavningar som bildar riktiga ord) i texter skrivna av andraspråksinlärare är svårt att detektera automatiskt. Vi har använt tre olika angreppssätt för granskningen: manuellt konstruerade feldetekteringsregler, statistiska skillnader mellan korrekt och felaktig text, samt maskininlärning av specifika feltyper.

De tre metoderna har vi utvärderat på en korpus bestående av svenska uppsatser av andraspråksinlärare. Vi fann att metoderna upptäcker olika fel och därför kompletterar varandra väl.

1 Introduction

Language technology has a potential to play a major role in the process of learning a language. Until recently, the use of language technology in systems for language learning has been nearly nonexistent. However, this has not been the case with grammar checkers for second language learners

[*]E-mail {johnny,viggo,knutsson,jsh}@nada.kth.se
This work was done in the research project *CrossCheck – a grammar checker for second language writers of Swedish* funded by Vinnova and KTH between 2001 and 2004, see http://www.nada.kth.se/theory/projects/xcheck/

learning English (see e.g. Bolt (1992), Chen (1997), Park et al. (1997), Yazdani (1990)). The question if grammar checkers actually improve second language learners' language is still a question of debate (Chen 1997; Vernon 2000). In spite of this, we see the adaptation of grammar checking for Swedish to second language learners as a first step to put language technology in computer assisted language learning environments.

Designing a grammar checker for second language learners raises several questions. Are second language learners writing much worse (at a grammatical level) essays than for instance native writers? Are they violating the grammatical rules of Swedish in a significant different way? These questions are hard to answer without large scaled and fine-grained corpus studies. In the CrossCheck project we have developed a corpus of second language Swedish, both for studying the types of errors and testing our grammar checkers, and as a general resource for the second language learning research community (Lindberg and Eriksson 2004).

However, the results of our studies so far are that the native writers and second language have many error types in common. They may not make the same instances of errors, but the error types are in many cases the same. Both groups make for instance agreement errors, split compounds and violate verb chain patterns. Based on these observations, our starting point have been to not treat second language writers as a special group of writers according to error types. Competence errors made by a second language writer could identical to performance errors of another writer, and vice versa. To judge if an error is made because of lack of grammatical knowledge, or if it was made because of typing failures, is extremely hard to model for a computer program.

If we develop better methods for grammar checking in general, we will develop better methods for grammar checking for second language writers and vice versa. Therefore we started with the method for grammar checking that we already had, namely the grammar checker Granska (Domeij et al. 1999), a state of the art grammar checker based on manually constructed error detection rules.

Our first question was therefore, how is Granska working on texts written by second language writers? Pilot studies were made (Öhrman 2000; Knutsson et al. 2002) and the major problem seemed to be limited coverage of the errors. Granska detected only about 30 % of the errors. Would it then be possible to increase the coverage of Granska without changing the method and technology used? The answer is no. The rule database of Granska was fine tuned with several hundreds of hours of work. If the goal is to increase coverage, the precision of Granska must also be altered. This means lower precision (more false alarms), and we did not believe that it

should be suitable for a user group containing language learners.

Since the error types are too many and too unpredictable we looked for new methods for grammar checking. We developed two statistical methods, ProbGranska and SnålGranska.

ProbGranska (Bigert and Knutsson 2002) detects errors by looking for grammatical constructions that are ”different” from known correct text. It detects improbable language constructs using part-of-speech tag trigram frequencies.

SnålGranska (Sjöbergh and Knutsson 2004) requires no manual work, only unannotated text and a few basic NLP tools. The method used is to annotate a lot of errors in written text and train an off-the-shelf machine learning implementation to recognize such errors. To avoid manual annotation artificially created errors are used for training.

In the following sections we will describe these three approaches to grammar checking and show how they perform individually and together on second language learner Swedish texts from the CrossCheck corpus, comparing the results to a commercial Swedish grammar checker.

2 The CrossCheck Learner Corpus

The CrossCheck Learner Corpus (or the SVANTE – SVenska ANdraspråks-TExter – Corpus) is a corpus of written second language Swedish. This is a kind of material that has been lacking for Swedish, where the emphasis long has been on the collection and transcription of spoken learner material, such as the EALA/ESFSLD Swedish component[1], a corpus containing spoken conversations and monologues of bilingual school children (Viberg 2001), the ASU (Andraspråkets StrukturUtveckling) Corpus (Hammarberg 1997), and possibly some others. Among these corpora, only ASU has a written component (about 1/3 of the total). Swedish is somewhat unique in this respect, since it is often remarked in the literature on English learner corpora that spoken learner materials are so scarce in comparison to written learner language corpora (e.g. Granger 1998a).

Like ICLE (the International Corpus of Learner English; Granger 1998b; Granger, Hung, and Petch-Tyson 2002) and ASU, the SVANTE Corpus also includes a native speaker part, argumentative essays written as part of Swedish high school national examinations.

A deliberate design feature of the corpus is that it is not intended to be ”balanced”, at least not by the way we compile it. Rather, we include as much material as we can lay our hands on, including as much relevant

[1] See `http://www.mpi.nl/ISLE/overview/Overview_ESFSLD.html`

metadata as possible, so that users will be able at anytime to extract ”virtual corpora” out of the material on the basis of the metadata.

3 Granska – A grammar checker using manually constructed rules

For several years we have developed Granska, a spelling and grammar checker for Swedish (Domeij et al. 1999). Granska consists of a spelling checker (Domeij et al. 1994), a part-of-speech tagger (Carlberger and Kann 1999), and about 350 manually constructed rules written in an object-oriented rule language constructed especially for Granska. About half of the rules are error detection and correction rules. An example of a rule detecting agreement errors in a noun phrase is the following.

```
cong22@incongruence {
   X(wordcl=dt),
   Y(wordcl=jj)*,
   Z(wordcl=nn &
     (gender!=X.gender | num!=X.num | spec!=X.spec))
-->
   mark(X Y Z)
   corr(X.form(gender:=Z.gender, num:=Z.num, spec:=Z.spec))
   info("The determiner" X.text
        "does not agree with the noun" Z.text)
   action(scrutinizing)
}
```

The first part of the rule detects the agreement error. The second part tells what should happen after a matching. The `mark` statement specifies that the erroneous phrase should be marked in the text, the `corr` statement that a function is used to generate a new inflection of the article from the lexicon, one that agrees with the noun. This correction suggestion is presented to the user together with a diagnostic comment (in the `info` statement) describing the error.

The rules are compiled and optimized using statistics of words and tag bigrams in Swedish. This means that each rule is checked by the matcher *only* at the positions in the text where the words or tag bigrams of the least probable position in the rule occur.

A subset of the rules of Granska constitutes a shallow parser for Swedish, called GTA (Knutsson et al. 2003).

4 ProbGranska – A statistically based grammar checker

ProbGranska is a probabilistic algorithm for detection of context-sensitive spelling errors (Bigert and Knutsson 2002). The algorithm is divided into two parts: a statistical part and a transformation part.

4.1 Statistical information

The first, statistical part of the algorithm uses PoS tag trigram frequency information, gathered from a corpus of the target language. The general observation is that a grammatical construction is probably malformed if it contains previously unseen trigrams. Unfortunately, human language is very productive, and new, unseen grammatical constructs will arise. To address this problem, we broaden the concept of a PoS tag trigram.

Two PoS tags that are used in similar syntactic contexts are said to be close. We want to use this closeness, or *distance*, between tags to mitigate the effect of rare trigrams due to the productivity of the language.

We calculate the distance between two tags by using the frequencies of PoS tag trigrams obtained from the corpus. Given two tags t and r, we look up the frequencies for the trigrams (t_1, t, t_2) and (t_1, r, t_2). Naturally, if either t or r is more frequent than the other, the trigram frequencies will be higher and thus, we have to compensate for the tag frequencies. We obtain $P_t(t_1, t_2) = \mathit{freq}(t_1, t, t_2)/\mathit{freq}(t)$.

From this, we can apply a number of similarity measures from the work of Lee (1999), e.g. the L1 norm: $L1(P_t, P_r) = \sum_{t_1,t_2} |P_t(t_1, t_2) - P_r(t_1, t_2)|$, where the sum is over all tag pairs t_1, t_2 in the tag set. We see that the distance increases when the trigrams differ in frequencies. The measure will give us a list of similarities between every pair of tags t and r. Suitably normalized the values can be used as probabilities. Thus, if p is $L1(P_t, P_r)$ normalized (i.e. the distance between t and r), p can be seen as the probability of retaining grammaticality when replacing a word having PoS tag t with a word having PoS tag r.

Now, given a rare trigram (t_1, t, t_2), we attempt to replace one (or more) of the tags with another tag close in distance. For example, if replacing t with r, we obtain (t_1, r, t_2). To penalize the tag change, we multiply the frequency of the new trigram with the probability p of the tag change. Now, if the penalized frequency is high (as defined by an arbitrary threshold e), the grammatical construction is most probably correct and the low frequency was originally due to a rare tag. If all attempted tag replacements result in low frequencies, the trigram is probably not grammatical and is marked as an error.

4.2 Phrase transformations

Most false alarms occur near the beginning or end of a phrase constituent. There, a trigram covers two phrases and the productivity of the language gives rise to almost any combination of PoS tags. Furthermore, rare phrase constructions often produce rare PoS tag trigrams. Normally, the simplest (or shortest) form of a phrase is the most common, e.g. *the men* is a more common type of NP than *the little green men*. Thus, when faced with a potential error as described in the previous subsection, we will identify all adjacent phrases and try to simplify. Hopefully, the rare trigram is due to a rare combination of phrases.

We attempt to transform a phrase to a simpler form by replacing it with a more common phrase of the same type. To this end, we use GTA, the shallow parser of Granska. Since we try to retain the inflectional information of the phrase, the new sentence will most probably be grammatical. For example: an error is detected near the words *are old are* in *All paintings that are old are for sale*. The NP *all paintings that are old* is reduced to *the paintings* and the sentence becomes *The paintings are for sale*, avoiding the rare construction.

The sentence resulting from the phrase transformation is fed to the algorithm in the previous section. If the new sentence is not erroneous, it is probably grammatically correct. If all phrase transformations fail (are reported as errors), there is probably a grammatical error and this is reported to the user. Rare PoS tag trigrams also occur frequently near a clause beginning or end. We decided not to look for errors in trigrams that cross a clause boundary. Hence, the largest unit is not a sentence but a clause.

4.3 Evaluation

The procedure used to evaluate the error detection algorithm is fully automated and requires no resources annotated with errors (Bigert 2004). Furthermore, it is portable to any language and tag set (given a dictionary in that language) and produces reproducible evaluations. The idea behind the procedure is to introduce artificial spelling errors into error-free text. A graph of precision versus recall is shown in Figure 1 where 2% errors were introduced into the text. As seen from the figure, the proposed method increases the precision significantly while sacrificing recall.

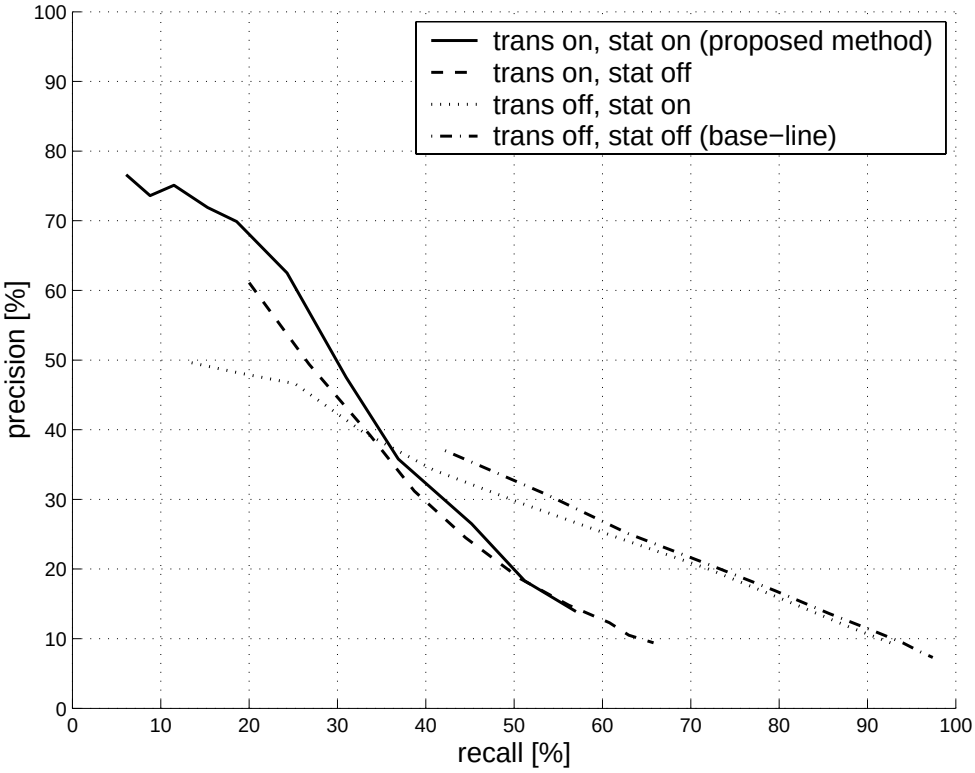

Figure 1: *Precision and recall of ProbGranska at the 2% error level.*

5 SnålGranska – A grammar checker requiring no manual work

The third method, SnålGranska (Stingy Checker), is based on machine learning of automatically constructed errors (Sjöbergh and Knutsson 2004). The main strength of SnålGranska is that almost no manual work is required to create it. It also has modest requirements on the NLP tools it uses. Another advantage is that the method is based on how correct language use looks, though not as much as ProbGranska. This means that it is able to detect errors that are hard to describe by manually written rules.

Currently, statistical methods are used for a wide range of tasks in the NLP area. One way to use this approach for grammar checking would be to treat it as a tagging task. First, collect a lot of text. Annotate all errors with ”ERROR” and the correct text with ”OK”. When this is done, train an off-the-shelf machine learning implementation on this annotated data. It will now be trained to detect errors in text.

This approach has two drawbacks. These methods generally require quite a lot of data, and finding enough text with unintentional errors in could be a problem. The largest drawback though is that a lot of manual work is required to find all errors and annotate them.

SnålGranska avoids this problem by using artificial errors. First, a lot of unannotated and (mostly) correct text is collected. Then the text is cor-

rupted by inserting simple errors automatically. Since these errors are inserted automatically, they can be annotated automatically at the same time. Then a machine learning algorithm is trained on this data and applied to new texts as a grammar checker.

The artificial errors that are generated can be of any type. As the strength of SnålGranska is the minimal amount of manual work required, only very simple errors have been used. Two errors types have been tested, split compound errors and agreement errors. Split compounds is an example of an error type that SnålGranska is well suited for. It is easy to split compounds, but the resulting sentence structure from split compounds is in general hard to predict, so it is hard to write manual rules for these errors. We use a modified spelling checker (Domeij et al. 1994; Kann et al. 2001) and split all compounds recognized by the spelling checker.

Agreement errors is an error type which SnålGranska is less suited for. It is the most thoroughly covered error type in current grammar checkers for Swedish, and it is relatively straightforward to write good rules for these manually. To see how well the SnålGranska method works compared to state of the art grammar checking, this was also tested. Errors were generated by randomly choosing words with gender, number or definiteness features, and replacing them with another word with the same lemma but another surface form, using a simple dictionary lookup.

Using more ”human like” artificial errors should be expected to produce a better grammar checker, but requires more work. If a lot of time is spent on producing a sophisticated error generation program, this time could perhaps have been spent better by writing rules for a traditional grammar checker. As long as the resulting grammar checker is useful, the simpler the error generation the better. For the examples above, about 15 minutes have been spent on error generation. This is the only manual work required, everything else is done automatically.

Almost any machine learning algorithm could be used for SnålGranska. We use fnTBL (Ngai and Florian 2001), which produces rules that are easily understood by humans. As features for the machine learner we use words and their part-of-speech, which is automatically assigned by a tagger. For split compounds we also use the spelling checker to filter out false alarms, by checking if the (suspected) errors combine into acceptable words. No other resources are needed.

When an artificial error results in a sentence that is also correct, which is quite possible, it will still be an error to the machine learning algorithm. This is not a problem, since there is also a lot of examples of correct language use in the training data. This means that (generally) only the properties of those artificially inserted errors that result in sentences that are not

	MS Word	Granska	Prob-Granska	Snål-Granska	Any Granska	Total
All detected errors	392	411	102	121	528	592
All false positives	21	13	19	19	48	–
Detected spelling errors	334	293	35	26	314	363
False positives	18	5	–	–	5	–
Detected gram. errors	58	118	67	95	214	229
False positives	3	8	19	19	43	–

Table 1: Evaluation on second language learner essays, 10 000 words. Any Granska means all errors detected by any of the Granska methods.

correct will be learned.

SnålGranska could be used on many error types, but so far only these two have been tested. This means that many easily detectable error types, such as repeated words or wrong verb tense after the infinitive marker *att*, are ignored by SnålGranska, which leads to low overall recall.

6 Evaluation

To evaluate the different grammar checking methods we used essays written by people learning Swedish as a second language. These were taken from the SSM-corpus part of the CrossCheck corpus. These texts contain a lot of errors, which is generally good for the grammar checkers (easier to get high precision), but it also leads to problems for the grammar checkers. Many errors overlap, which can give unexpected results. There is also often very little correct text to base any analysis on.

All methods were run on about 10 000 words of text from the essays. The grammar checker for Swedish in Microsoft Word 2000 was also run on the same text, as a comparison to other available methods. The grammar checker in MS Word has been developed for high precision, while for instance SnålGranska was developed for high recall (on the two error types it detects). All alarms from the grammar checkers were manually checked to see if there was a true error or a false alarm. Results are shown in Table 1. The texts were not manually checked to find all errors, but a manually checked sample shows that many errors go undetected. Less than half of the errors in the sample were detected.

The grammar checkers using manually constructed rules, Granska and MS Word, show higher precision (about 95%) than the other methods (about 85%). They also detect many more errors, mainly because they also look for spelling errors, which are common and much easier to detect. When it

Pair		Both	Only Granska	Only Prob-Granska	Only Snål-Granska	Any
Granska+ProbGr.	Correct	17	101	50		168
	False alarms	0	8	19		27
Granska+SnålGr.	Correct	44	74		51	169
	False alarms	3	5		16	24
ProbGr.+SnålGr.	Correct	11		56	84	151
	False alarms	0		19	19	38

Table 2: Pairwise overlap in detection of grammatical errors between Granska, ProbGranska and SnålGranska.

comes to grammatical errors the recall is similar for all methods.

While the manual rules of Granska detect more errors, and with higher precision, than the other methods, it still misses many errors detected by other methods. ProbGranska in particular was developed explicitly to find errors which are hard to detect using manual rules. In Table 2 the pairwise overlap in detections of grammatical errors for the different methods developed in the CrossCheck project is shown. The three methods complement each other and by combining them much better coverage can be achieved.

Even SnålGranska, which currently is only trained on split compound errors and agreement errors, two error types already covered by Granska, finds many errors that the other methods do not detect. It could also be extended with more error types, for improved recall.

Combining the different methods could be done in many ways. One simple method is to treat any detection from any method as an error. In Table 1 the results using this method are shown.

7 Discussion

The evaluation of our three approaches to grammar checking in Section 6 showed that the three methods to a large extent detect different errors and therefore complement each other well. We therefore propose that a grammar checker should combine different approaches to grammar checking.

How can a grammar checker be further improved to detect even more of the errors? All three methods described in this chapter rely on the same type of part-of-speech disambiguation. The main problem is that grammatical errors sometimes are misinterpreted as correct grammatical constructions. Independent of which grammar checking method that is used after word class disambiguation, many errors cannot be detected. Our initial studies

showed that word class disambiguation is necessary to limit the amount of false alarms. What we need is a language model that is much more rigid than the current model. This is a veritable case of Heller's Catch 22, a rigid language model would not analyse ill-formed constructions at all, and we are thereby back into the deep parsing dilemma – where many sentences are not parsed either because they are ungrammatical or because of limitations of the current grammar. The problem of the general analysis of ungrammatical constructions is one of the main bottlenecks for further improvements of current methods for grammar checking.

The methods for grammar checking described in this chapter are already integrated (Granska and ProbGranska) or close to be integrated (SnålGranska) into a language-learning environment called Grim[2]. Grim is a web client with basic word processing facilities, which is connected to several network based language tools (bilingual lexicons, a grammatical analyzer, a word inflector, a interface to a concordancer).

In the design of Grim it has been important to provide the user with several different views of language. For the case of grammar checking, Granska represents a rule-based view of language, ProbGranska a more statistical view, and SnålGranska is something in between. The idea and the contribution of using three methods for grammar checking, beside increased coverage and accuracy, is to make the user aware of how different tools can give different feedback on the user's writing, and that different linguistic resources will treat language in different ways. One pedagogical problem is how to explain for the user that three methods are better than one. A second pedagogical problem is how to show that the three methods co-exist in an environment with several language tools seamlessly integrated. One important kind of feedback that we have got from the users of Grim so far, is that they view Grim as one program, not as an interface to several different language tools and programs.

Acknowledgements

We are grateful to Lars Borin, Janne Lindberg and Gunnar Erikssson for their work on the CrossCheck corpus, and to Stefan Westlund, who developed the Grim interface to Granska and ProbGranska.

[2]See `http://skrutten.nada.kth.se`

References

Bigert, J. 2004. Probabilistic detection of context-sensitive spelling errors. In *Proc. 4th Int. Conf. Language Resources and Evaluation (LREC 2004)*.

Bigert, J. and Knutsson, O. 2002. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In *Proc. 2nd Workshop of Robust Methods in Analysis of Natural Language Data*, pp. 10–19.

Bolt, P. 1992. An evaluation of grammar-checking programs as self-help learning aids for learners of English as a foreign language. *Computer Assisted Learning 5*(1–2), 49–91.

Carlberger, J. and Kann, V. 1999. Implementing an efficient part-of-speech tagger. *Software–Practice and Experience 29*(9), 815–832.

Chen, J. F. 1997. Computer generated error feedback and writing process: A link. *TESL-EJ Teaching English as a second Foreign Language 2*(3).

Domeij, R., Hollman, J., and Kann, V. 1994. Detection of spelling errors in Swedish not using a word list en clair. *J. Quantitative Linguistics 1*, 195–201.

Domeij, R., Knutsson, O., Carlberger, J., and Kann, V. 1999. Granska – an efficient hybrid system for Swedish grammar checking. In *Proc. 12th Nordic Conf. on Computational Linguistics*.

Granger, S. 1998a. The computer learner corpus: A versatile new source of data for SLA research. In S. Granger (Ed.), *Learner English on computer*, pp. 3–18. London: Longman.

Granger, S. (Ed.) 1998b. *Learner English on Computer*. London: Longman.

Granger, S., Hung, J., and Petch-Tyson, S. (Eds.) 2002. *Computer learner corpora, second language acquisition and foreign language teaching*. Number 6 in Language Learning and Language Teaching. Amsterdam: John Benjamins.

Hammarberg, B. 1997. *Manual of the ASU corpus, a longitudinal text corpus of adult learner Swedish. Version 1997–04–10*. Stockholm University, Department of Linguistics.

Kann, V., Domeij, R., Hollman, J., and Tillenius, M. 2001. Implementation aspects and applications of a spelling correction algorithm. In

L. Uhlirova, G. Wimmer, G. Altmann, and R. Koehler (Eds.), *Text as a Linguistic Paradigm: Levels, Constituents, Constructs. Festschrift in honour of Ludek Hrebicek*, Volume 60 of *Quantitative Linguistics*, pp. 108–123. Trier, Germany: WVT. Available on the web from `http://www.nada.kth.se/theory/projects/swedish.html`.

Knutsson, O., Bigert, J., and Kann, V. 2003. A robust shallow parser for Swedish. In *Proc. 14th Nordic Conf. on Computational Linguistics*.

Knutsson, O., Pargman, T. C., and Eklundh, K. S. 2002. Computer support for second language learners' free text production – Initial studies. In *Proc. 5th Int. Workshop on Interactive Computer Aided Learning*.

Lee, L. 1999. Measures of distributional similarity. In *Proc. 37th Annual Meeting of the ACL*, pp. 25–32.

Lindberg, J. and Eriksson, G. 2004. CrossCheck-korpusen – en elektronisk svensk inlärarkorpus. In *Proc. ASLA 2004 Conference*.

Ngai, G. and Florian, R. 2001. Transformation-based learning in the fast lane. In *Proceedings of NAACL-2001*, Carnegie Mellon University, Pittsburgh, USA, pp. 40–47.

Park, J. C., Palmer, M., and Washburn, G. 1997. An English grammar checker as a writing aid for students of English as a second language. In *Proc Conf. on Applied Natural Language Process.*

Sjöbergh, J. and Knutsson, O. 2004. Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. In preparation.

Vernon, A. 2000. Computerized grammar checkers 2000: capabilities, limitations, and pedagogical possibilities. *Computers and Composition 17*, 329–349.

Viberg, Å. 2001. Age-related and L2-related features in bilingual narrative development in Sweden. In L. Verhoeven and S. Strömqvist (Eds.), *Narrative development in a multilingual context*, pp. 87–128. Amsterdam: John Benjamins.

Yazdani, M. 1990. An artificial intelligence approach to second language learning. *J. Artificial Intelligence in Education 1*, 85–90.

Öhrman, L. 2000. Datorstödd språkgranskning och andraspråksinlärare. Technical report, Institutionen för lingvistik, Stockholms Universitet. D-uppsats i datorlingvistik.