Leveraging Native Data to Correct Preposition Errors in Learners' Dutch

Lennart Kloppenburg and Malvina Nissim
Center for Language and Cognition Groningen – Language Technology
Rijksuniversiteit Groningen, The Netherlands
l.a.m.kloppenburg@student.rug.nl, m.nissim@rug.nl

Abstract

We address the task of automatically correcting preposition errors in learners' Dutch by modelling preposition usage in native language. Specifically, we build two models exploiting a large corpus of Dutch. The first is a binary model for detecting whether a preposition should be used at all in a given position or not. The second is a multiclass model for selecting the appropriate preposition in case one should be used. The models are tested on native as well as learners data. For the latter we exploit a crowdsourcing strategy to elicit native judgements. On native test data the models perform very well, showing that we can model preposition usage appropriately. However, the evaluation on learners' data shows that while detecting that a given preposition is wrong is doable reasonably well, detecting the absence of a preposition is a lot more difficult. Observing such results and the data we deal with, we envisage various ways of improving performance, and report them in the final section of this article.

Keywords: Preposition correction, language modelling, learners' data, Dutch

1. Introduction

Computer-assisted language learning is a field where language technology is combined with human language acquisition. To make it possible, we need systems that are able to recognise errors and suggest corrections. However, in spite of recent campaigns on automatic error correction (Ng et al., 2013; Ng et al., 2014), and in spite of some notable exceptions (Nicholls, 2003) the lack of large amounts of error-annotated data remains a bottleneck in the development and evaluation of systems that support language learning, especially for L2s other than English.

One obvious solution is creating error-aware resources to build and test robust models of learners' language use (Han et al., 2010), which has however substantial costs, especially in terms of human effort. Another way of bypassing this shortage of data is to exploit *native* data, which can be considered as having gold labels of correct usage of language (Han et al., 2006; Tetreault and Chodorow, 2008; Gamon et al., 2008; De Felice and Pulman, 2009). This latter strategy is what we adopt in this work, focusing on preposition errors in learners of Dutch. As far as we are aware, this is the first such work for this language.

It has been noted that in many languages prepositional constructions are the most difficult for foreign language learners, because of their versatile and often ambiguous character (Dale et al., 2012). Although we have no specific figures on this kind of error in learners of Dutch, we know that prepositions are used extremely frequently and that large corpora of native Dutch are available, which can be exploited to build models of correct preposition usage.

In this paper we describe a two-stage approach to the detection and correction of preposition errors in essays written by learners of Dutch. We train two models on native data that are to be deployed in a pipeline. The models are tested both on withheld native data as well as learners' data. For the latter, we crowdsourced correctedness judgments from native speakers, as the corpus we drew test data from is not specifically error-annotated.

2. Phenomenon and Task

In Dutch, there are just over a hundred prepositions, and only some of them are used commonly. In the `cdb` (Alpino Treebank, (van Noord, 2006)), a corpus of ca. 140,000 words from the Eindhoven corpus (newspaper text), about 15% of all tokens are prepositions, despite only representing 0.66% of all types (*n*=15,863), with the top 15 prepositions comprising almost 90% of all preposition use (see Table 1 for details). In other words, prepositions are used very frequently despite being a relatively small lexical class.

Table 1: Frequency distribution of the 15 most Dutch prepositions in the `cdb` (van Noord, 2006). The total number of prepositions in the corpus is 18,790.

Preposition	Translation	Frequency
van	of	22.8%
in	in	16.3%
op	on	7.2%
te	at	7.1%
voor	for/before	6.3%
met	with	5.7%
aan	on, to	4.3%
door	through, by	3.3%
bij	at, with	2.8%
uit	out	2.6%
om	by, around	2.6%
over	over, about	2.4%
tot	until, to	2.2%
naar	to	2.1%
als	if, as	1.2%
total		88.9%
others		11.1%

The problem of automatically correcting preposition errors in texts written by language learners should be divided into two subtasks. The first is *error detection*, i.e. train a system to spot errors. The second is *error correction*, the task of providing the language learner with appropriate feedback or correction. Three basic situations of preposition misuse can be identified (Eeg-Olofsson and Knutsson, 2003; Chodorow et al., 2007; Liu, 2008):

1. insertion (a preposition is invoked erroneously)
2. deletion (a preposition is omitted erroneously)
3. substitution (a preposition is picked erroneously)

Deletion and insertion errors are shown in Examples (1) and (2), respectively. In (a) we report the learner's sentence with the error, and in (b) the correct version.

(1) a. *Hij zet tot 8 soorten **van** dranken, maar geen chocolade melk. *(It can make up to eight types **of** drinks, but no hot chocolate.)*

b. Hij zet tot 8 soorten drank, maar geen chocolademelk.

(2) a. *Roger Federer werd **ø** 1981 geboren. *(*Roger Federer was born **ø** 1981.)*

b. Roger Federer werd **in** 1981 geboren.

In a substitution error, as in Example (3), the learner understands that a preposition must be used, but picks the wrong one (a). The task is to find out this out, and replace the preposition with the correct one (b).

(3) a. *Vanaf nu gaat alle communicatie **door** email, en niet de telefoon. *(From now on, all communication will be done **through** email, rather than the telephone.)*

b. Vanaf nu gaat alle communicatie **via** email, en niet de telefoon.

This classification effectively separates distinct types of errors which might require different strategies to solve. Note that the *deletion* and *insertion* errors are mirrored, as shown in Table 2, hinting at a possibly joint treatment.

Table 2: Mirroring of *deletion* and *insertion* error types.

	Correct use	**Learner sample**
Deletion	prep *P*	ø
Insertion	ø	prep *P*

Indeed, for *insertion* and *deletion*, the task is to model presence or absence of a preposition in any given context, and could be thus conceived as a binary classification task.

3. Data

To build and test our models we used two datasets. The first is LASSY Large (LArge Scale SYntactic Annotation of Written Dutch (van Noord et al., 2011; Van Noord et al., 2013)), a syntactically annotated corpus of native Dutch which includes newswire and Wikipedia articles, for a total of 700M words and over 64M sentences.

The second is Leerdercorpus Nederlands (Perrez and Degand, 2009), an as of yet relatively unused corpus which contains 3,468 essays and argumentative texts written by students who study Dutch as a second language, with a variety of mother tongues. The total amount of tokens is reported at 774,658. The distribution of mother languages expresses quite a bit of variation, including French (1247), German (877), Polish (599), Hungarian (413), Indonesian (197), English (9) and others which are not defined further (125), making it not quite possible to rely on the regularities of source-language-specific error types.

To the best of our knowledge, this is the only learner corpus for Dutch, but it is not error-annotated, meaning that the density of preposition errors is unknown and that evaluation cannot be done automatically against a gold standard (see Section 5.2. for further details).

4. Method

In order to detect potential preposition errors, one needs to be able to discriminate between good language use and bad language use. In absence of error-annotated data, we trained models of correct preposition usage on LASSY Large, with the assumption that native data provides gold labels on grammatical choices. Building on the observations on error types in Section 2., we trained two different models. The first is a *binary* detection model trained on positive and negative preposition events, and used to predict whether a given context contains a preposition or not. The second is a *multiclass* selection model: a classifier with fifteen prepositions as class values, used to select a preposition if none was present (as indicated by the detection model), or if the present preposition was wrong. They are deployed in a pipeline.

4.1. Detection model

Using SVM, we trained a model to detect insertion and deletion errors: given a feature-vector, the model predicts whether this vector is built around a white-space (absence) or a preposition (presence). We define a white-space vector as a vector which is extracted around a case which had no preposition, but could potentially harbour a preposition given its surrounding linguistic properties. We determined this by comparing POS n-gram chunks without prepositions with POS bigram windows on either side of a preposition. To give an example, let us consider a case where we have a POS pattern that includes a preposition: *VERB_ADV_PREP_NOUN_$* (where $ is end of sentence). Any case with the same POS context *but without a preposition*, such as *VERB_ADV_NOUN_$*, can be considered as a *non-trivial case of preposition absence*. A pairwise example to illustrate the concept of this *non-trivial* preposition absence is given in (4) and (5).

(4) De man gaat vaak **naar** concerten. *(The man goes **to** concerts often.)*

(5) De man bezoek vaak **ø** concerten. *(The man visits **ø** concerts often.)*

Note how in the first sentence, the verb "gaat" (*goes*) asks for a preposition, in this case "naar" (*to*). The second sentence is written without a preposition but has exactly the same surrounding structure in terms of parts of speech. The difference here is that "bezoek" (*visits*) does not require a preposition as it is a transitive verb and takes a direct object. As explained above, since the surrounding POS pattern could hypothetically hold a preposition (as in Example (4)), we call this a case of non-trivial preposition absence. Because we are in fact talking about the *absence* of a preposition, these have to be considered as *negative* instances and their (preposition) class value is set to "false". A preposition case, i.e. a *positive* instance with class value "true", is a vector whose linguistic properties were extracted around a preposition. The actual task is to then train the model to find distinctions between these two types of vectors.

The model is informed by n-gram-based features (see Table 3), both at the word- and at the POS-level, which define the linguistic context in which the preposition or the whitespace occurs.

Table 3: Features for the detection model.

Feature	Description
bigram_left	Token bigram left of preposition
bigram_right	Token bigram right of preposition
trigram_left	Token trigram left of preposition
trigram_right	Token trigram left of preposition
bigram_postags_left	POS bigram left of preposition
bigram_postags_right	POS bigram right of preposition
trigram_postags_left	POS trigram left of preposition
trigram_postags_right	POS trigram right of preposition

A feature selection experiment based on information gain run on a small dataset of 150K examples, showed that surrounding POS N-grams are highly indicative in discerning between preposition absence or presence. We include an overview of the most informative features in Table 4.

The possible outcomes of the model are shown in Table 5.

4.2. Selection model

The selection model is used for selecting suitable prepositions. As outlined in Table 5, this can occur in two situations: because of a *substitution* error or because of a *deletion* error. Note that before feeding the vector with a preposition to this selection model, it is still unknown whether this preposition is correct or incorrect. Attesting this is done by analysing the output from the model. Table 6 illustrates the different outcomes of the selection model.

For this task, we selected the 15 most frequent prepositions in Dutch (see Table 1), and trained an SVM model on vectors of native data, using the features in Table 7, inspired by (Chodorow et al., 2007). The selected preposition is predicted as a single-label classification task.

5. Results

The detection and selection models were trained on 2M and 20M feature vectors, respectively. The models were subsequently tested on 268,895 feature vectors of unseen native data and on 1,499 items of the L2 data.

5.1. Baselines

In order to have a lower bound to compare our system to, we devised a few baselines for each model, with increasing predictive power.

Table 4: Feature-selection results for the detection model based on Information Gain.

InfoGain	Feature	Value
0.242	bigram_postags_right	*verb_det*
0.211	trigram_postags_right	*verb_det_noun*
0.177	bigram_postags_right	*det_noun*
0.054	bigram_postags_right	*det_name*
0.052	bigram_postags_right	*det_adj*
0.048	trigram_postags_right	*det_noun_punct*
0.048	trigram_postags_right	*det_adj_noun*
0.048	trigram_postags_right	*adv_det_noun*
0.048	bigram_postags_right	*adv_det*
0.046	trigram_postags_right	*name_name_punct*
0.046	bigram_postags_left	*det_noun*
0.046	bigram_postags_right	*name_name*
0.044	bigram_postags_right	*noun_det*

Table 5: Outcomes and procedure for the detection model.

Observed	Predicted	Message	Action
Presence	Presence	No error (yet)	Feed to *selection model*
Presence	Absence	*Insertion error*	Delete preposition
Absence	Presence	*Deletion error*	Feed to *selection model*
Absence	Absence	No error	Stop.

Table 6: Outcomes and procedure for the selection model. *Assigned by the detection model.

Observed	Predicted	Message	Action
None	P_j	Deletion error*	Correction: use P_j
P_j	P_j	No error	None
P_j	$P_{\neg j}$	Substitution error	Correction: use $P_{\neg j}$

Table 7: Feature set for the substitution model.

Feature	Description
bigram_left	Token bigram left of preposition
bigram_right	Token bigram right of preposition
trigram_left	Token trigram left of preposition
trigram_right	Token trigram left of preposition
PRE_verb	Verb preceding the preposition
FOLL_verb	Verb following the preposition
PRE_noun	Noun preceding the preposition
FOLL_noun	Noun following the preposition
bigram_postags_left	POS bigram left of preposition
bigram_postags_right	POS bigram right of preposition
trigram_postags_left	POS trigram left of preposition
trigram_postags_right	POS trigram right of preposition
FOLL_phr_head	Headword of following phrase
PRE_phr	Preceding phrase-type
FOLL_phr	Following phrase-type
Preposition	The preposition

Detection model Baseline 1 is a most-frequent-class simple model always predicting no preposition for any given context. Baseline 2 is based on the 200 most common POS patterns for preposition presence cases: if a certain pattern has been observed in training data as more likely to occur in presence of a preposition rather than in its absence, this baseline will predict a preposition. Baseline 3 is as Baseline 2, but *all* POS patterns are considered. In both cases, POS patterns are based on trigrams.

Selection model Baseline 1 is always predicting the preposition *van* (most frequent). Baseline 2 is a basic classifier trained on the surrounding POS bigrams only.

5.2. Evaluation on native data

Table 8 outlines the results (including results for the baselines) on native test data for the detection model, which only determines either the absence or the presence of a preposition. The results for the detection model are surprisingly high (but so are those for Baseline 3). This might be due to the fact that negative preposition vectors are all extracted in the same way (as explained in Section 4.1.). In other words, it could depend also on the way the data is represented. Future work will investigate this further, especially in terms of what exactly makes a non-trivial whitespace case.

Table 8: Detection model's and baselines' performance on native test data.

	Precision	Recall	F-score
Detection Model	1.00	1.00	1.00
Baseline 1	0.25	0.50	0.33
Baseline 2	0.75	0.51	0.36
Baseline 3	0.97	0.97	0.97

Table 9 shows the results on native test data for the selection model. A comparison with the baselines is in Table 10.

Table 9: Selection model's performance on native test data with breakdown per preposition.

Preposition	Prec	Rec	F	# cases
aan	0.72	0.69	0.71	11,690
als	0.81	0.51	0.63	5,157
bij	0.57	0.43	0.49	8,369
door	0.66	0.55	0.60	9,154
in	0.72	0.81	0.76	47,649
met	0.67	0.67	0.67	17,167
naar	0.69	0.62	0.65	6,232
om	0.77	0.69	0.73	10,881
op	0.78	0.78	0.78	23,111
over	0.65	0.59	0.62	6,036
te	0.97	0.97	0.97	19,746
tot	0.75	0.68	0.71	6,703
uit	0.69	0.56	0.62	6,514
van	0.80	0.87	0.83	71,102
voor	0.64	0.58	0.61	19,380
Average	0.75	0.75	0.75	*Total*: 268,895

As we can see from the tables, results show that the system is very accurate for some prepositions (*van*, *in*, *te*, *op*), while not so much for others (*als*, *bij*, *door*). Most prepositions with low scores seem to occur less than the more easily predicted prepositions (*als* and *bij*), though this is not always the case (*voor*, *met*). Conversely, the system performs quite well for the preposition *tot*, which is also quite infrequent.

We assume that learner data will contain more erroneous spelling but less variety in structure because of the fact that most data from the Dutch Learning Corpus consists of student essays. Erroneous spelling, poor lexical choice and grammatical errors will likely contribute to the results on learner test data.

Table 10: Comparison of the performance of the selection model and the baselines on native test data.

	Precision	Recall	F-score
Selection Model	0.75	0.75	0.75
Baseline 1	0.07	0.26	0.11
Baseline 2	0.38	0.42	0.35

5.3. Evaluation on learners' data

Because the learners' corpus is not error-annotated we collect human judgements via crowdsourcing. Annotators were asked to assess whether any given spot would contain a preposition, and in case which one. More than one choice was allowed, so that a system's decision in the selection model is deemed as correct if the predicted preposition is included in the set of those chosen by the annotators.

In order to gather annotations through crowdsourcing, we developed a web application that allowed users to create an account and annotate sentences written by L2 students, directly extracted by the Leerdercorpus Nederlands. Sentences of any length or structure were matched for extraction. An important reason to refrain from performing any preselection was to exploit the data in its actual form and distribution. For every sentence, a preposition (or whitespace) was replaced by a question mark and participants were asked to select prepositions (from the fifteen prepositions that were included in the model) that would fit the context or indicate that the context should not have a preposition at all. Figure 1 shows a screenshot of the annotation interface.

The resulting annotated subset of the learners' data consists of 1,499 cases. Of these, 971 were based on actual prepositions and were picked randomly from the system's decisions, while ensuring that every preposition would be represented. The rest were attempts of the system to detect deletion errors, so that the presence of a preposition is the system's guess. In the test set, the annotators identified 105 errors (7%). Of these: 72 were substitution errors, 29 insertion errors and 4 deletion errors, as shown in Table 11.

We also show the confusion matrix for the selection model in Table 12. Apart from the encouraging diagonal line showing the correct decisions, we cannot observe any specific confusion patterns. We hypothesise that this can be due to the varied nature of the mother tongues represented in the Leerdercorpus Nederlands, making it unlikely for any regular error trend to arise. A more focused analysis on a single mother tongue might show a different picture in this sense, but is left to future work.

6. Discussion and Outlook

With an F1-score of 75% on L1 data, and results generally above the baselines, we show that it is possible to build a fairly accurate model of fifteen Dutch prepositions. The considerably lower results on L2 data indicate that the differences between native and learner language are substan-

Figure 1: Screenshot of the web-based annotation interface used for crowdsourcing native judgements on sentences extracted from the Leerdercorpus Nederlands.

Table 11: Results on learners' data. We do not report performance on correction for the baselines, as they have not been deployed in a pipeline but only applied separately.

	Detection		Correction	
	Prec	**Rec**	**Accuracy**	**N (gold)**
Substitution	0.14	0.90	0.60	72
Insertion	0.22	0.21	–	29
Deletion	0.01	0.75	1.00	4
Correct	0.99	0.36	–	1,394
Average/Total	0.91	0.38	0.62	1,499
Baseline 1	0.14	0.37	–	–
Baseline 2	0.39	0.37	–	–
Baseline 3	0.43	0.55	–	–

Table 12: Confusion matrix for model selection (columns) and gold standard annotation (rows) in learners' test data.

	aan	**als**	**bij**	**door**	**in**	**met**	**naar**	**om**	**op**	**over**	**te**	**tot**	**uit**	**van**	**voor**
aan	**24**	0	2	2	4	7	0	1	0		0	0	1	0	4
als	1	**1**	2	1	3	0	0	1	1	2	0	1	0	0	2
bij	2	0	**14**	2	9	3	0	1	4	0	0	0	0	5	2
door	1	0	3	**2**	3	2	0	3	1	1	0	0	2	0	1
in	2	0	5	0	**117**	4	0	1	5	0	0	0	2	5	3
met	0	0	5	1	15	**41**	1	7	2	3	0	0	0	6	1
naar	1	0	1	0	2	1	**12**	0	2	0	0	0	0	2	0
om	0	0	1	0	8	3	1	**35**	0	2	0	1	0	2	1
op	1	1	1	0	12	0	3	1	**48**	1	0	0	1	2	2
over	1	0	2	1	6	3	0	3	1	**20**	0	0	0	0	1
te	1	0	0	0	2	0	0	0	1	4	**87**	0	0	2	2
tot	0	0	0	1	0	1	0	3	3	0	0	**5**	0	0	0
uit	0	0	2	0	1	0	0	0	0	0	0	1	**7**	0	1
van	2	0	2	2	15	6	1	2	3	3	0	0	0	**93**	8
voor	5	0	8	3	8	12	0	8	3	0	2	0	1	4	**70**

tial, especially because native data does not provide any information on the types of errors and confusions that learners make (Han et al., 2010). The detection model suffers mostly from the transition from L1 to L2 test data. The model is clearly too biased towards preposition presence, which makes little sense, as the chances that any given whitespace contains a preposition are intuitively very low. Insertion error detection and correction are both rather low, yet promising. Most insertion errors were confused as substitution errors, another consequence of the detection model because it would rather see a preposition than none. The high recall for substitution is encouraging, and the low precision is worrying; they are both due to the fact that the selection model picks one preposition only, even if other candidates might be similarly acceptable.

In this sense, one logical next step is rebuilding the selection procedure into a *multilabel* model, which would reduce the number of false positives, and also improve the learner's experience by suggesting more than one option to a (likely) wrongly used preposition. Multiple suggestions would reduce the number of false positives when detecting substitution errors (De Felice and Pulman, 2009). Adding more information to the model, such as semantic features relative to word classes, for example, could also yield better results. For example, features could be inherited by similar words, so as to cope with unseen nouns or verbs. To this end, similarity information could be exploited either in the form of simple synonym lookup via lexical resources such as WordNet (Fellbaum, 1998), or via corpus-derived distributional information in the form, for example, of word embeddings (Mikolov et al., 2013).

However, a prime direction for improvement is to make the model more familiar with L2 language. One simple first step could just be using a spell corrector to improve the quality of L2 data and overall preprocessing. Linguistically-informed features on learners' use of prepositions should also be investigated, though the variety of mother tongues in the L2 corpus doesn't make this straightforward. Another interesting possibility to be investigated in this context would be to add artificially generated errors to the native training corpora (as suggested in (Rozovskaya and Roth, 2010)), so as to reduce the differences between native and learner language. Specifically, Rozovskaya and Roth (2010) show that introducing artificial errors in article usage in English in the training data increases results on learner test data. They make sure that the distribution of these errors is similar to the distribution of errors in ESL (English as a Second Language) data. This could prove very useful as it teaches the system what kind of errors can be expected and which errors are rare, and might be especially interesting for the case of prepositions as they are a closed class but with many possible different confusions.

As far as the corpus itself is concerned, the LCN corpus proved to be useful and promising for this task because it features typical learner errors. There is much more data in the corpus which we did not annotate in the short amount of time we had. A next step for the LCN would be to make it error-annotated so tasks on automatic error correction or language acquisition can be tackled more conveniently.

Acknowledgements

We would like to sincerely thank Liesbeth Degand for providing us with the Leerdercorpus Nederlands, and those who partook in annotating the L2 data. We are also grateful to the reviewers for their helpful comments, which have undoubtably improved the quality of this paper.

References

Chodorow, M., Tetreault, J. R., and Han, N.-R. (2007). Detection of grammatical errors involving prepositions. *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, pages 25–30, June.

Dale, R., Anisimoff, I., and Narroway, G. (2012). Hoo 2012: A report on the preposition and determiner error correction shared task. *The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 54–62, June.

De Felice, R. and Pulman, S. (2009). Automatic detection of preposition errors in learner writing. *CALICO Journal*, pages 512–528.

Eeg-Olofsson, J. and Knutsson, O. (2003). Automatic grammar checking for second language learners the use of prepositions. In *Proceedings of Nodalida 2003*.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Gamon, M., Gao, J., Brockett, C., Klementiev, A., Dolan, W. B., Belenko, D., and Vanderwende, L. (2008). Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, pages 449–456.

Han, N., Chodorow, M., and Leacock, C. (2006). Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129.

Han, N., Tetreault, J., hwa Lee, S., and young Ha, J. (2010). Using an error-annotated learner corpus to develop and esl/efl error correction system. In *Prcoeedings of LREC 2010*.

Liu, Y. (2008). The effects of error feedback in second language writing. *Arizona Working Papers in SLA & Teaching*, 15:65–79.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the ICLR Workshop*, pages 1–12.

Ng, H. T., Wu, S. M., Wu, Y., Hadiwinoto, C., and Tetreault, J. (2013). The conll-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland, June. Association for Computational Linguistics.

Nicholls, D. (2003). The Cambridge learner corpus—Error coding and analysis for lexicography. In D. Archer, et al., editors, *Proceedings of the Corpus Linguistics 2003 Conference*, pages 572–581, Lancaster, UK. UCREL, Lancaster University.

Perrez, J. and Degand, L. (2009). Het Leerdercorpus Nederlands in theorie en in de praktijk. *IVN - 17e Colloquium Neerlandicum*.

Rozovskaya, A. and Roth, D. (2010). Training paradigms for correcting errors in grammar and usage. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 154–162, Los Angeles, California, June. Association for Computational Linguistics.

Tetreault, J. R. and Chodorow, M. (2008). The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 865–872, Stroudsburg, PA, USA. Association for Computational Linguistics.

van Noord, G., Schuurman, I., and Bouma, G. (2011). Handleiding lassy syntactische annotatie. Technical report, University of Groningen, September.

Van Noord, G., Bouma, G., Van Eynde, F., De Kok, D., Van der Linde, J., Schuurman, I., Sang, E. T. K., and Vandeghinste, V. (2013). Large scale syntactic annotation of written Dutch: Lassy. In *Essential Speech and Language Technology for Dutch*, pages 147–164. Springer.

van Noord, G. (2006). At Last Parsing Is Now Operational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.