ISCA Archive
http://www.isca-speech.org/archive

Speech and Language Technology in Education (SLaTE2011)
Venice, Italy
August 24-26, 2011

Improving ASR processing of ungrammatical utterances through grammatical error modeling

Helmer Strik, Joost van Doremalen, Janneke van de Loo, Catia Cucchiarini

Centre for Language and Speech Technology, Radboud University, Nijmegen, The Netherlands

{h.strik;j.vandoremalen;c.cucchiarini}@let.ru.nl; jannekevandeloo@student.ru.nl

Abstract

Automatic speech recognition (ASR) of non-native utterances with grammatical errors is problematic. A new method which makes it possible to better recognize such utterances is presented in the current paper. It can be briefly summarized as follows: extract error patterns automatically from a learner corpus, formulate rewrite rules for these syntactic and morphological errors, build finite state grammars (FSGs), and use these FSGs as language models in ASR systems. All rules used in isolation and in different combinations yield lower word error rates (WERs).

Index Terms: computer-assisted language learning (CALL), non-native speech, grammatical errors, automatic speech recognition (ASR), language modeling

1. Introduction

ASR-based 'computer assisted language learning' (CALL) systems for second language (L2) learning are, by definition, intended for people who do not yet speak the target language properly and are thus likely to make errors, including grammatical errors. Especially these grammatical errors may constitute an obstacle to ASR processing of the learners' utterances, while a CALL system should in principle be able to recognize erroneous learners' spoken output in order to proceed to error detection and possibly provide corrective feedback.

So far, little attention has been paid to how language models (LMs) in ASR systems should be adapted to improve ASR of ungrammatical utterances, although there are some exceptions. For instance, [6] used N-grams. By adding non-native data to the training material a small decrease in WER (from 52.0% to 47.8%) was achieved. Another approach based on FSGs appeared to be more successful [1, 5, 11].

For this reason we also decided to use FSGs in our research. While in previous studies FSGs were often handcrafted and could only handle deletions and substitutions, our goal is to generate LMs automatically and to obtain a more comprehensive modeling of grammatical errors that also includes transpositions (Tp) and insertions (Ins), since these errors also occur in L2 speech. Part of this research has been reported on in [7]. In the present paper we focus on the processing of transpositions, and on the use of a syntactic parser for this purpose. Here we can only briefly describe the method, results, etc. For more details see [8].

So, the problem we faced in our research was how to improve ASR of ungrammatical utterances. We had a corpus of non-native utterances and thought of applying NLP tools such as a POS tagger [9, 12] or a syntactic parser [10, 13] to analyze them. However, it turned out that these NLP tools generally do not perform satisfactorily on ungrammatical utterances. Therefore, we decided to adopt a different method which performs a comparison of target utterances and realized utterances to extract information on L2 grammatical errors, which is subsequently employed to formulate rules for the language model in the ASR. The present research is carried out in the framework of the DISCO project [3, 14], and the resulting method is currently used in the FASOP project [15].

In this paper we first present the method and material used for this research (Section 2). Section 3 describes the results concerning the error patterns and the ASR performance research. These results are discussed in Section 4.

2. Material and method

2.1. Material

We made use of existing L2 speech material taken from the Dutch JASMIN speech corpus [2]. Recordings were made for DL2 (Dutch as a second language) speakers with many different mother tongues who had relatively low proficiency levels, namely A1, A2 and B1 of the Common European Framework (CEF) [16]. For the experiments reported on in this paper we used the extemporaneous speech contained in the JASMIN human-machine dialogues. This corpus comes with orthographic transcriptions that were manually created and include (dis-)fluency phenomena such as filled pauses, restarts and repetitions. Grammatical errors were manually annotated at a later stage by two trained annotators. In addition to syntactic and morphological errors, these utterances also contain other errors that may concern the pronunciation of individual sounds, prosody and disfluencies. Some of these utterances are (very) difficult to understand even for human listeners. After removal of very short utterances of only 1 or 2 words, the dataset consists of 2088 utterances.

2.2. Method

The method consists of 4 steps:

a) For every utterance (*realization*) the corresponding correct (target) utterance is obtained
b) Automatic error analysis
c) Rule formulation and implementation
d) ASR experiment

The corresponding correct (target) utterance (in step a) was obtained manually by interpreting what the speaker said and then correcting the form of the utterance. This task was carried out by two trained annotators. In this task, they had access to the speech itself, the orthographic transcription of the utterance and the context in which the utterance was made. The annotators were instructed to keep the corrected form as close to the realized utterance as possible.

Steps b and d are carried out automatically, and consist of the following stages:

b) Error analysis:

1. For the target utterances: Part-of-speech (POS) tagging using Tadpole [9, 12] & syntactic parsing using Alpino [10, 13].
2. Alignment of the words in the target and realized utterances.
3. Matching of the words in target and realized utterances.
4. Listing of error patterns in terms of transformations of POS tags and parse (sub)trees.

d) ASR experiments:

1. On the basis of the target utterances and error rules, lists of candidate utterances (including the error types) are generated.
2. Language models are generated based on the lists of candidate utterances in the form of an unweighted FSG.
3. An ASR system [4, 8] utilizing these language models yields the recognized utterance.
4. The recognized utterances are matched with the lists of candidate utterances and the errors are retrieved.
5. Feedback is provided to the user.

In this paper we investigate the resulting error patterns (Step b) and evaluate the performance of the ASR (Step d) by calculating the word error rate (WER).

3. Results

3.1. Error patterns

Many grammatical errors were found which differ considerably in terms of absolute (Abs.) and relative (Rel.) frequency (see Table 1). Since both magnitudes are important, we decided to order the errors according to the product of these two measures: ARF = Abs. x Rel. Frequency. Table 1 shows that ARF rapidly diminishes. The majority of deletion errors concern the deletion of articles (rule no. 1), which is a well known phenomenon in DL2 speech. Substitution errors generally appear to be related to morphological errors. These errors can be easily formulated in terms of POS tags.

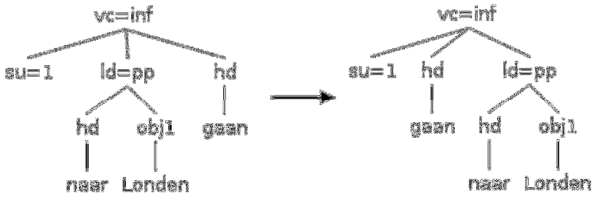

Figure 1: Example of a parse tree transformation. See Section 3.1 for details.

Transposition rules 4 and 7 represent errors related to the position of verbs. For example, rule 4 describes a transposition of the finite verb in a subordinate clause (from clause end to post-subject position). In general, rules related to transpositions and insertions can be described better using a hierarchical syntactical representation of the utterance (instead of a 'flat' POS representation). For this reason, we have described a number of these phenomena using transformations of parse trees. An example of such a phenomenon is the position of the infinite verb in a verb clause, such as in 'ik wil naar Londen gaan' ('I want to go to Londen'), which is often incorrectly uttered as 'ik wil gaan naar Londen'. In Fig. 1, this is represented as a transformation of a parse tree. Another example is the position of the finite verb in a sentence starting with an adverbial clause. In this case, the position of the subject and that of the verb are inversed. An example is 'Elk jaar gaan *veel toeristen* naar Istanbul' ('Every year *a lot of tourists* go to Istanbul') which is often incorrectly realized without inversion: 'Elk jaar *veel toeristen* gaan naar Istanbul' (See Fig. 2).

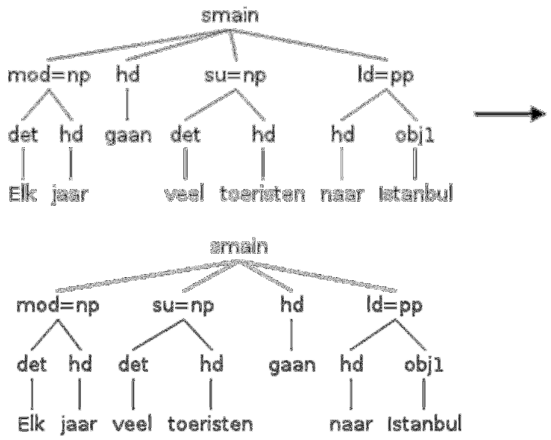

Figure 2: Example of a parse tree transformation. See Section 3.1 for details.

3.2. ASR performance

The results of the ASR experiments are listed in Table 2. The five most frequent deletion rules, the five most frequent substitution rules and the two most frequent transposition rules (POS-based and parse-tree based) are used first in isolation and then in different combinations. All added rules and combinations of rules cause the system to perform significantly better ($p<.05$) than a baseline system in which language models containing only the correct target utterance are used. The deletion rules are the most beneficial (27.5%), before the transposition rules (POS-based: 27.9% and parse-tree based: 27.6%) and the substitution rules (28.2%). The deletion and substitution rules combined perform still better (27.3%). Adding either the POS-based or parse-tree based transposition rules increases the performance even more, to 26.6% and 26.3% respectively. Thus, the parse-tree based implementations of the transposition rules yield lower WERs, compared to POS-based implementations.

4. Discussion and conclusions

Our automatic analysis procedure was capable of finding many grammatical errors. Some of these errors had already been described in the literature, but clearly not all of them. In addition to providing information on not yet described errors, our data-driven method also produced quantitative information on the frequency of occurrence of these errors.

The results show that rules in isolation and in different combinations yield lower WERs. One could thus wonder whether the WER can be further reduced by adding more extra rules. However, experiments we conducted with a word loop as the language model produced higher WERs in the order of 50-60%. In other words, allowing too much freedom leads to higher WERs.

Table 1. Results of the analysis procedure

Results are ranked according to ARF = Absolute (Abs.) X Relative (Rel.) Frequency. In Column 2 the type of the error rules are listed (Del = deletion, Sub = substitution, Tp = transposition). Rules are formatted in the following way: POS:a → b where the left hand side a is rewritten as the right hand side b, POS represents the POS tag and a represents the word (: any word; Ø: no word).*

No.	Type	Rule	Frequency		
			Abs.	Rel.	ARF
1	Del	Article: * → Ø	316	43.5 %	137.54
2	Sub	Possessive Pronoun: 'mijn' → 'mij'	104	68.4 %	71.61
3	Del	Personal Pronoun: * → Ø	207	18.3 %	37.82
4	Tp	Subordinator: * + Nominative Personal Pronoun: * + * + Finite Verb: * → Subordinator: * + Nominative Personal Pronoun: * + Finite Verb: * + *	61	58.1 %	35.44
5	Sub	Adjective with final 'e': * → Adjective without final 'e'	74	31.6 %	23.38
6	Sub	Article: 'het' → 'de'	46	29.3 %	13.48
7	Tp	Finite Verb + Nominative Personal Pronoun → Nominative Personal Pronoun + Finite Verb	43	23.9 %	10.28
8	Del	Preposition: * → Ø	87	10.3 %	8.94
9	Del	Conjunction: 'dat' → Ø	20	42.6 %	8.51
10	Del	Adverb: 'er' → Ø	18	40.0 %	7.20
11	Sub	Adjective without final 'e': * → Adjective with final 'e'	64	10.4 %	6.67
12	Del	Finite Verb: * → Ø	89	7.1 %	6.30
13	Sub	Plural Noun: * → Singular Noun	38	12.6 %	4.79
14	Sub	Interrogative without final 'e': * → Interrogative with final 'e'	4	100 %	4.00

Table 2. Results of the ASR experiments

In the rows:

Row 2: Base line results
Rows 3-5: Del and Sub rules
Rows: 6-9: Tp POS rules
Rows 10-13: Tp parser rules

In the columns:

Column 1: Rules used to generate the language models.
For columns 2 to 7 (where # = number of words):
- Sub: substitutions, Del: deletions, Ins: insertions,
- #Err: number of errors in ASR results = #Sub + #Del + #Ins,
*- WER: Word Error Rate = 100% * #Err /#words (=15367),*
- ΔErr: actual decrease in #Err, compared to baseline = #Err(baseline) - #Err = 4450 - #Err,
- ΔErr_{max}: maximum reduction in #Err possible, if the best path in the FSG is always chosen,
*- Ratio: percentage of decrease in #Err actually accomplished = 100% * $\Delta Err/\Delta Err_{max}$*

Condition	#Sub	#Del	#Ins	#Err	WER	ΔErr	ΔErr_{max}	Ratio
Baseline (only target)	1454	1852	1144	4450	29.0%	-	47	-
5 Del rules	1332	2282	615	4229	27.5%	221	726	30.4%
5 Sub rules	1333	1856	1144	4333	28.2%	117	322	36.3%
5 Del + 5 Sub	1269	2309	624	4202	27.3%	248	993	25.0%
Tp Rule 4 - POS	1395	1814	1106	4315	28.1%	135	148	91.2%
Tp Rule 7 - POS	1393	1842	1134	4369	28.4%	81	118	68.6%
2 Tp rules - POS	1381	1805	1097	4283	27.9%	167	219	76.3%
2 Tp - POS + 5 Del + 5 Sub	1258	2236	586	4080	26.6%	370	1142	32.4%
Tp Rule 4 – parser	1388	1805	1097	4290	27.9%	160	174	92.0%
Tp Rule 7 – parser	1400	1835	1127	4362	28.3%	88	124	71.0%
2 Tp rules – parser	1378	1789	1080	4247	27.6%	203	253	80.2%
2 Tp - parser + 5 Del + 5 Sub	1261	2223	564	4048	26.3%	402	1188	33.8%

So far, we have used the rules of the most frequent errors. It is possible that the reduction in WER gradually decreases as we add more rules (of less frequent errors) and maybe that a turning point is reached where WER starts to increase again. Further research is needed to investigate whether there is such a turning point, and, if so, where. In other words, how many rules should be used and which ones.

In the current method the analysis leading to the error patterns and the LM generation were performed automatically, while rule formulation and implementation was done partly manually. However, rule formulation could also be carried out automatically, for instance by using machine learning techniques [see e.g. 11]. This would make it easier to study more rules and their effect as well as optimal rule selection. The problem is though whether the (mainly) linguistic knowledge we used in our current approach to formulate the rules can easily be applied in automatic rule formulation procedures.

We also found lower WERs by using transposition and insertion rules. The results of the insertion rules are not presented here; they turned out to be less important, being less frequent, and have less effect on the WER. However, good results were obtained with the transposition rules, compared to those produced by deletion and insertion rules.

The ratios presented in Table 2 vary from 30-92%, indicating that the optimal path in the FSG is often chosen, but not always. There could be several reasons for this. The current acoustic models were trained on native speech and could be improved, e.g. by training them on non-native speech, and using speaker adaptation. Another possibility is to include prior probabilities in the FSGs.

It can also be observed in Table 2 that the ratios are higher for transposition rules. This is probably due to the fact that the transposition rules cause more dissimilar paths, for which also the acoustics differ more. Put otherwise, it seems logical that a transposition of words is easier to detect by ASR systems and humans (in making the annotations), and that they agree on it, than a subtle difference regarding the presence (or not) of a 'small' phoneme (e.g. /@/ or /n/). The transposition is also categorical, it is either present or not, while this is not the case for the phonemes. The reduction of phonemes is gradual, from slightly and extremely, to completely, and it is known that ASR systems and human listeners often do not agree on these issues. So, it is also an evaluation problem. Another evaluation problem is that besides the grammatical errors there are many other 'interfering factors', e.g. a lot of disfluencies, noise, incomprehensible words, etc., which increases the WERs.

We often noticed that NLP tools in general perform less satisfactorily on ungrammatical utterances. It turned out though that the Tadpole tagger [9, 12] is more robust in this sense than the Alpino parser [10, 13]. So the need for deriving target utterances during analysis is higher for the Alpino parser than for the Tadpole tagger.

Another reason for using target utterances during analysis is that in a CALL system the target utterances are often available from the start, one generally knows what the language learner should say. The rules can then be applied to generate a language model which makes it possible to better recognize the learners' ungrammatical utterances.

Our research shows that using a parser has certain advantages compared to using a POS tagger. Some errors can be described in a (linguistically) more plausible, effective way using a parser. Furthermore, we obtained better results (higher ratios, lower WERS) for the transposition rules using the Alpino parser (compared to the Tadpole tagger). However, an advantage of POS tagging is that it is less complex, easier to use during analysis and LM generation, and also that it is more robust to ungrammatical utterances. Therefore, in our approach we used POS tags where possible, and only in cases where POS tagging was not sufficient did we make use of a parser. This might be a good rule of thumb.

Finally, we would like to point out that although this method was applied here to L2 speech, it can in principle be used for native speech as well, and not only for Dutch, but also for other languages.

5. Acknowledgements

The current research was conducted within the framework of the DISCO project, carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (http://taalunieversum.org/taal/technologie/stevin/).

6. References

Papers in alphabetical order:

[1] Bernstein, J., Najmi, A., Ehsani, F., "Subarashii: Encounters in Japanese spoken language education", CALICO, 16, 361–384, 1999.

[2] Cucchiarini, C., van Doremalen, J. and Strik, H., "DISCO: Development and Integration of Speech technology into Courseware for language learning", In Proceedings of Interspeech, pp. 2791-2794, 2008.

[3] Cucchiarini, C., Driesen, J., Van Hamme, H. and Sanders, E. (2008) "Recording speech of children, non-natives and elderly people for HLT applications: the JASMIN-CGN corpus", Proceedings of LREC-2008.

[4] Demuynck, K., Roelens, J., Van Compernolle, D., and Wambacq, P., "SPRAAK: An Open Source SPeech Recognition and Automatic Annotation Kit". In Proceedings of Interspeech, pp. 495-498, 2008.

[5] Kweon, O.P., Ito, A., Suzuki, M., Makino, S., "A grammatical error detection method for dialog-based CALL system", Journal of Natural Language Processing, 12(4), 137-156, 2005.

[6] Raux, A., & Eskenazi, M., "Using task-oriented spoken dialogue systems for language learning: Potential practical applications and challenges", Proc. ISCA ITRW INSTiL04, pp. 147–150. Venice, Italy, 2004.

[7] Strik, H., van de Loo, J.,van Doremalen, J. and Cucchiarini, C., "Practicing Syntax in Spoken Interaction: Automatic Detection of Syntactic Errors in Non-Native Utterances", In Proceedings of SLaTE, 2010.

[8] Van de Loo, J., "Language modeling for ASR-based CALL applications", Master's Thesis, Radboud University Nijmegen, 2011 (http://lands.let.ru.nl/~strik/publications/theses/van-de-Loo-Ma-thesis.pdf)

[9] Van den Bosch, A., Busser, G.J., Daelemans, W. and Canisius, S. (2007) "An efficient memory-based morphosyntactic tagger and parser for Dutch", in F. van Eynde, P. Dirix, I. Schuurman, and V. Vandeghinste (Eds.), , pp. 99-114, Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting, Leuven, Belgium, 2007.

[10] Van Noord, G. (2006). At Last Parsing Is Now Operational. In: P. Mertens, C. Fairon, A. Dister, P. Watrin (Eds.), TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles, pp. 20-42.

[11] Wang, H. and Kawahara, T., "Effective error prediction using decision tree for ASR grammar network in CALL system", In Proceedings of ICASSP, 2008.

[12] http://ilk.uvt.nl/tadpole/

[13] http://www.let.rug.nl/vannoord/alp/Alpino/

[14] http://lands.let.ru.nl/~strik/research/DISCO/

[15] http://lands.let.ru.nl/~strik/research/FASOP.html

[16] http://www.webcef.eu/?q=node/121

.