Realistic Grammar Error Simulation using Markov Logic

Sungjin Lee
Pohang University of Science and Technology
Pohang, Korea
junion@postech.ac.kr

Gary Geunbae Lee
Pohang University of Science and Technology
Pohang, Korea
gblee@postech.ac.kr

Abstract

The development of Dialog-Based Computer-Assisted Language Learning (DB-CALL) systems requires research on the simulation of language learners. This paper presents a new method for generation of grammar errors, an important part of the language learner simulator. Realistic errors are generated via Markov Logic, which provides an effective way to merge a statistical approach with expert knowledge about the grammar error characteristics of language learners. Results suggest that the distribution of simulated grammar errors generated by the proposed model is similar to that of real learners. Human judges also gave consistently close judgments on the quality of the real and simulated grammar errors.

1 Introduction

Second Language Acquisition (SLA) researchers have claimed that feedback provided during conversational interaction facilitates the acquisition process. Thus, interest in developing Dialog-Based Computer Assisted Language Learning (DB-CALL) systems is rapidly increasing. However, developing DB-CALL systems takes a long time and entails a high cost in collecting learners' data. Also, evaluating the systems is not a trivial task because it requires numerous language learners with a wide range of proficiency levels as subjects.

While previous studies have considered user simulation in the development and evaluation of spoken dialog systems (Schatzmann et al., 2006), they have not yet simulated grammar errors because those systems were assumed to be used by native speakers, who normally produce few grammar errors in utterances. However, as telephone-based information access systems become more commonly available to the general public, the inability to deal with non-native speakers is becoming a serious limitation since, at least for some applications, (e.g. tourist information, legal/social advice) non-native speakers represent a significant portion of the everyday user population. Thus, (Raux and Eskenazi, 2004) conducted a study on adaptation of spoken dialog systems to non-native users. In particular, DB-CALL systems should obviously deal with grammar errors because language learners naturally commit numerous grammar errors. Thus grammar error simulation should be embedded in the user simulation for the development and evaluation of such systems.

In Foster's (2007) pioneering work, she described a procedure which automatically introduces frequently occurring grammatical errors into sentences to make ungrammatical training data for a robust parser. However the algorithm cannot be directly applied to grammar error generation for language learner simulation for several reasons. First, it either introduces one error per sentence or none, regardless of how many words of the sentence are likely to generate errors. Second, it determines which type of error it will create only by relying on the relative frequencies of error types and their relevant parts of speech. This, however, can result in unrealistic errors. As exemplified in Table 1, when the algorithm tries to create an error by deleting a word, it would probably omit the word 'go' because verb is one of the most frequent parts of speech omitted resulting in an unrealistic error like the first simulated output. However, Korean/Japanese language learners of English tend to make subject-verb agreement errors, omission errors of the preposition of prepositional verbs, and omission errors of articles because their first language does not have similar grammar rules so that they may be slow on the uptake of such constructs. Thus, they often commit errors like the second simulated output.

Input sentence
He wants to go to a movie theater
Unrealistic simulated output
He wants to to a movie theater
Realistic simulated output
He want go to movie theater

Table 1: Examples of simulated outputs

This paper develops an approach to statistical grammar error simulation that can incorporate this type of knowledge about language learners' error characteristics and shows that it does indeed result in realistic grammar errors. The approach is based on Markov logic, a representation language that combines probabilistic graphical models and first-order logic (Richardson and Domingos, 2006). Markov logic enables concise specification of very complex models. Efficient open-source Markov logic learning and inference algorithms were used to implement our solution.

We begin by describing the overall process of grammar error simulation and then briefly reviewing the necessary background in Markov logic. We then describe our Markov Logic Network (MLN) for grammar error simulation. Finally, we present our experiments and results.

2 Overall process of grammar error simulation

The task of grammar error simulation is to generate an ill-formed sentence when given a well-formed input sentence. The generation procedure involves three steps: 1) Generating probability over error types for each word of the well-formed input sentence through MLN inference 2) Determining an error type by sampling the generated probability for each word 3) Creating an ill-formed output sentence by realizing the chosen error types (Figure 1).

3 Markov Logic

Markov logic is a probabilistic extension of finite first-order logic (Richardson and Domingos, 2006). An MLN is a set of weighted first-order clauses. Together with a set of constants, it defines a Markov network with one node per ground atom and one feature per ground clause. The weight of a feature is the weight of the first-order clause that originated it. The probability of a state x in such a network is given by $P(x) = (1/Z)\,exp\,(\sum_i w_i f_i(x))$, where Z is a normalization constant, w_i is the weight of the ith clause, $f_i = 1$ if the ith clause is true, and $f_i = 0$ otherwise.

Markov logic makes it possible to compactly specify probability distributions over complex relational domains. We used the learning and inference algorithms provided in the open-source Alchemy package (Kok et al., 2006). In particular, we performed inference using the belief propagation algorithm (Pearl, 1988), and generative weight learning.

4 An MLN for Grammar Error Simulation

This section presents our MLN implementation which consists of three components: 1) Basic formulas based on parts of speech, which are comparable to Foster's method 2) Analytic formulas drawn from expert knowledge obtained by error analysis on a learner corpus 3) Error limiting formulas that penalize statistical model's over-generation of nonsense errors.

4.1 Basic formulas

Error patterns obtained by error analysis, which might capture a lack or an over-generalization of knowledge of a particular construction, cannot explain every error that learners commit. Because an error can take the form of a performance slip which can randomly occur due to carelessness or tiredness, more general formulas are needed as a default case. The basic formulas are represented by the simple rule:

• $PosTag(s, i, +pt) \Rightarrow ErrorType(s, i, +et)$

where all free variables are implicitly universally quantified. The "$+pt, +et$" notation signifies that the MLN contains an instance of this rule for each (part of speech, error type) pair. The evi-

		He	wants	to	go	to	a	movie	theater	
										1 step
Inference	*v_agr_sub*	0.000	**0.371**	0.000	0.000	0.000	0.000	0.000	0.000	
	prp_lex_del	0.000	0.000	**0.284**	0.000	0.269	0.000	0.000	0.000	
	at_del	0.000	0.000	0.000	0.000	0.000	**0.355**	0. 000	0.000	
	...	...	...	...	...	...	...	...	...	
	none	0.921	0.449	0.604	0.866	0.605	0.506	0.781	0.798	
Sampling		none	v_agr_sub	prp_lex_del	none	none	at_del	none	none	*2 step*
Realization		He	***want***		go	to		movie	theater	*3 step*

Figure 1: An example process of grammar error simulation

dence predicate in this case is $PosTag(s, i, pt)$, which is true iff the ith position of the sentence s has the part of speech pt. The query predicate is $ErrorType(s, i, et)$. It is true iff the ith position of the sentence s has the error type et, and inferring it returns the probability that the word at position i would commit an error of type et.

4.2 Analytic formulas

On top of the basic formulas, analytic formulas add concrete knowledge of realistic error characteristics of language learners. Error analysis and linguistic differences between the first language and the second language can identify various error sources for each error type. We roughly categorize the error sources into three groups for explanation: 1) Over-generalization of the rules of the second language 2) Lack of knowledge of some rules of the second language 3) Applying rules and forms of the first language into the second language.

Often, English learners commit pluralization error with irregular nouns. This is because they over-generalize the pluralization rule, i.e. attaching 's/es', so that they apply the rule even to irregular nouns such as 'fish' and 'feet' etc. This characteristic is captured by the simple formula:

• $IrregularPluralNoun(s, i) \wedge PosTag(s, i, NNS) \Rightarrow ErrorType(s, i, N\_NUM\_SUB)$

where $IrregularPluralNoun(s, i)$ is true iff the ith word of the sentence s is an irregular plural and N_NUM_SUB is the abbreviation for substitution by noun number error.

One trivial error caused by a lack of knowledge of the second language is using the singular noun form for weekly events:

• $Word(s, i-1, on) \wedge DayNoun(s, i) \wedge PosTag(s, i, NNS) \Rightarrow ErrorType(s, i, N\_NUM\_SUB)$

where $Word(s, i-1, on)$ is true iff the $i-1$th word is 'on' and $DayNoun(s, i)$ is true iff the ith word of the sentence s is a noun describing day like Sunday(s). Another example is use of plurals behind 'every' due to the ignorance that a noun modified by 'every' should be singular:

• $Word(s, di, every) \wedge DeterminerRel(s, di, ni) \Rightarrow ErrorType(s, ni, N\_NUM\_SUB)$

where $DeterminerRel(s, di, ni)$ is true iff the dith word is the determiner of the nith word.

An example of errors by applying the rules of the first language is that Korean/Japanese often allows omission of the subject of a sentence; thus, they easily commit the subject omission error. The following formula is for the case:

• $Subject(s, i) \Rightarrow ErrorType(s, i, N\_LXC\_DEL)$

where $Subject(s, i)$ is true iff the ith word is the subject and N_LXC_DEL is the abbreviation for deletion by noun lexis error.[1]

4.3 Error limiting formulas

A number of elementary formulas explicitly stated as hard formulas prevent the MLN from generating improbable errors that might result from over-generations of the statistical model. For example, a verb complement error should not have a probability at the words that are not complements of a verb:

• $!\,VerbComplement(s, vi, ci) \Rightarrow !\,ErrorType(s, ci, V\_CMP\_SUB).$

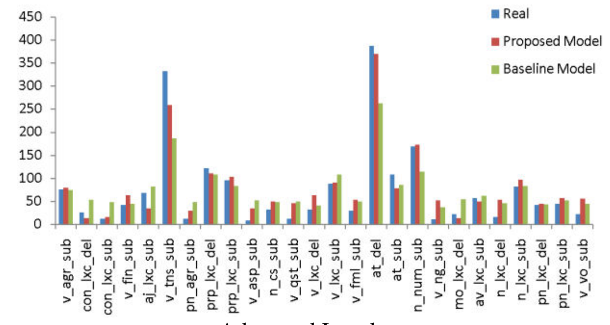
where "!" denotes logically 'not' and "." at the end signifies that it is a hard formula. Hard formulas are given maximum weight during inference. $VerbComplement(s, vi, ci)$ is true iff the cith word is a complement of the verb at the vith position and V_CMP_SUB is the abbreviation for substitution by verb complement error.

5 Experiments

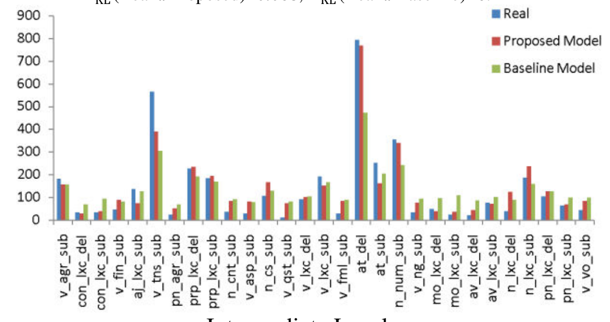
Experiments used the NICT JLE Corpus, which is speech samples from an English oral proficiency interview test, the ACTFL-ALC Standard Speaking Test (SST). 167 of the files are error annotated. The error tagset consists of 47 tags that are described in Izumi (2005). We appended structural type of errors (substitution, addition, deletion) to the original error types because structural type should be determined when creating an error. For example, V_TNS_SUB consists of the original error type V_TNS (verb tense) and structural type SUB (substitution). Level-specific language learner simulation was accomplished by dividing the 167 error annotated files into 3 level groups: Beginner(level1-4), Intermediate(level5-6), Advanced(level7-9).

The grammar error simulation was compared with real learners' errors and the baseline model using only basic formulas comparable to Foster's algorithm, with 10-fold cross validations performed for each group. The validation results were added together across the rounds to compare the number of simulated errors with the number of real errors. Error types that occurred less than 20 times were excluded to improve reliability. Result graphs suggest that the distribution of simulated grammar errors generated by the proposed model using all formulas is similar to that of real learners for all level groups and the

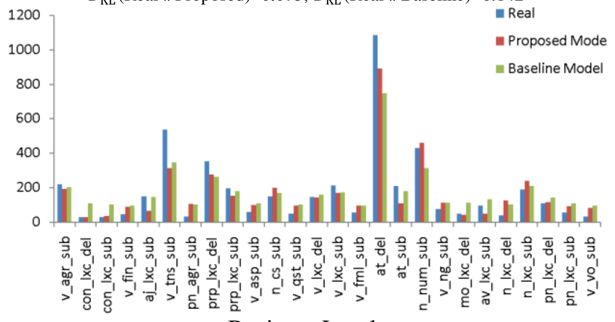
[1] Because space is limited, all formulas can be found at http://isoft.postech.ac.kr/ges/grm_err_sim.mln



Advanced Level:
D_{KL} (Real ‖ Proposed)=**0.068**, D_{KL} (Real ‖ Baseline)=0.122



Intermediate Level:
D_{KL} (Real ‖ Proposed)=**0.075**, D_{KL} (Real ‖ Baseline)=0.142



Beginner Level:
D_{KL} (Real ‖ Proposed)=**0.075**, D_{KL} (Real ‖ Baseline)=0.092

Figure 2: Comparison between the distributions of the real and simulated data

proposed model outperforms the baseline model using only the basic formulas. The Kullback-Leibler divergences, a measure of the difference between two probability distributions, were also measured for quantitative comparison. For all level groups, the Kullback-Leibler divergence of the proposed model from the real is less than that of the baseline model (Figure 2).

Two human judges verified the overall realism of the simulated errors. They evaluated 100 randomly chosen sentences consisting of 50 sentences each from the real and simulated data. The sequence of the test sentences was mixed so that the human judges did not know whether the source of the sentence was real or simulated. They evaluated sentences with a two-level scale (0: Unrealistic, 1: Realistic). The result shows that the inter evaluator agreement (kappa) is moderate and that both judges gave relatively close judgments on the quality of the real and simulated data (Table 2).

	Human 1	Human 2	Average	Kappa
Real	0.84	0.8	0.82	0.46
Simulated	0.8	0.8	0.8	0.5

Table 2: Human evaluation results

6 Summary and Future Work

This paper introduced a somewhat new research topic, grammar error simulation. Expert knowledge of error characteristics was imported to statistical modeling using Markov logic, which provides a theoretically sound way of encoding knowledge into probabilistic first order logic. Results indicate that our method can make an error distribution more similar to the real error distribution than the baseline and that the quality of simulated sentences is relatively close to that of real sentences in the judgment of human evaluators. Our future work includes adding more expert knowledge through error analysis to incrementally improve the performance. Furthermore, actual development and evaluation of a DB-CALL system will be arranged so that we may investigate how much the cost of collecting data and evaluation would be reduced by using language learner simulation.

Acknowledgement

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute for Information Technology Advancement) (IITA-2009-C1090-0902-0045).

References

Foster, J. 2007. Treebanks Gone Bad: Parser evaluation and retraining using a treebank of ungrammatical sentences. IJDAR, 10(3-4), 129-145.

Izumi, E et al. 2005. Error Annotation for Corpus of Japanese Learner English. In Proc. International Workshop on Linguistically Interpreted Corpora

Kok, S. et al. 2006. The Alchemy system for statistical relational AI. http://alchemy.cs.washington.edu/.

Pearl, J. 1988. Probabilistic Reasoning in Intelligent Systems Morgan Kaufmann.

Raux, A. and Eskenazi, M. 2004. Non-Native Users in the Let's Go!! Spoken Dialogue System: Dealing with Linguistic Mismatch, HLT/NAACL.

Richardson, M. and Domingos, P. 2006. Markov logic networks. Machine Learning, 62(1):107-136.

Schatzmann, J. et al. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies, The Knowledge Engineering Review, Vol. 21:2, 97–126