Developing Learner Corpus Annotation for Korean Particle Errors

Sun-Hee Lee
Wellesley College
slee6@wellesley.edu

Markus Dickinson
Indiana University
md7@indiana.edu

Ross Israel
Indiana University
raisrael@indiana.edu

Abstract

We aim to sufficiently define annotation for post-positional particle errors in L2 Korean writing, so that future work on automatic particle error detection can make progress. To achieve this goal, we outline the linguistic properties of Korean particles in learner data. Given the agglutinative nature of Korean and the range of functions of particles, this annotation effort involves issues such as defining the tokens and target forms.

1 Introduction and Motivation

One area of analyzing second language learner data is that of detecting errors in function words, e.g. prepositions, articles, and particles (e.g., Tetreault and Chodorow, 2008; De Felice and Pulman, 2008; de Ilarraza et al., 2008; Dickinson et al., 2011; Tetreault et al., 2010; Han et al., 2006), as these tend to be problematic for learners. This work has developed much, but it has mostly been for English. We thus aim to further the development of methods for detecting errors in functional elements across languages, by developing annotation for post-positional particles in Korean, a significant source of error for learners (Ko et al., 2004; Lee et al., 2009) and an area of interest for computer-assisted language learning (CALL) (Dickinson et al., 2008). As there is at present very little work on annotated learner corpora for morphologically-rich languages, this represents a significant step forward.

There have been some efforts for annotating particle errors in Korean, but they have not directly linked to automatic error detection. The corpus in Lee et al. (2009) is made up of college student essays; is divided according to student level (beginner, intermediate) and student background (heritage, non-heritage);[1] and is hand-annotated for particle errors. This corpus, however, does not contain gold standard segmentation, requiring users to semi-automatically determine particle boundaries. In addition to segmentation, to make particle error detection a widespread task where real systems are developed, we need to outline the scope of particle errors (e.g., error types, influence of other errors) and incorporate insights into an annotation scheme.

Selecting the correct particle in Korean is complicated by many factors. First, particles combine with preceding words in written Korean, as opposed to being set apart by white space, as in English. Thus, segmentation plays an integrated role. Secondly, selecting a particle for annotation is not a simple question, as they are sometimes optional, influenced by surrounding errors, and can be interchangeable. Thirdly, Korean particles have a wide range of functions, including modification and case-marking. Annotation, and by extension the task of particle error detection, must account for these issues.

We focus on the utility of annotation in evaluating particle error detection systems, ensuring that it can support the automatic task of predicting the correct particle (or no particle) in a given context. Given that other languages, such as Japanese and Arabic, face some of the same issues (e.g., Hanaoka et al., 2010; Abuhakema et al., 2008), fleshing them out for error annotation and detection is useful beyond this one situation and help in the overall process of "developing best practices for annotation and evalu-

[1] Heritage learners have had exposure to Korean at a young age, such as growing up with Korean spoken at home.

ation" of learner data (Tetreault et al., 2010).

2 Korean particles

Korean postpositional particles are morphemes[2] that appear after a nominal to indicate a range of linguistic functions, including grammatical functions, e.g., subject and object; semantic roles; and discourse functions. In (1), for instance, *ka* marks the subject (function) and agent (semantic role).

(1) Sumi-**ka** John-**uy** cip-**eyse** ku-**lul** twu
Sumi-SBJ John-GEN house-LOC he-OBJ two
sikan-**ul** kitaly-ess-ta.
hours-OBJ wait-PAST-END
'Sumi waited for John for (the whole) two hours in his house.'

Similar to English prepositions, particles can have modifier functions, adding meanings of time, location, instrument, possession, etc., also as in (1). Note here that *ul*/*lul* has multiple uses.[3]

Particles are one of the most frequent error types for Korean language learners (Ko et al., 2004).

3 Defining particle error annotation

3.1 Defining the tokens

Korean is agglutinative: words are generally formed by attaching suffixes to a stem. Particles are written without spaces, making token definitions non-trivial. In the next three sections, we discuss a three-layered annotation, where the output of one layer is used as the input for the next.

Spacing errors Given the differences in word formation and spacing conventions (e.g., compounds are often written without spaces), spacing errors are common for learners of Korean (Lee et al., 2009). As particles are word-final entities, correcting spacing errors is necessary to define where a particle can be predicted. This is similar to predicting a preposition between two words when those words have been merged. Consider (2). To see where the particle *-lul* is to be inserted, as in (2b), the original merged form in (2a) must be split.[4]

(2) a. O: yey-tul-myen
example-take-if
'if (we) take an example'

b. C: yey-lul tul-myen
example-OBJ take-if

We also correct words which have incorrectly been split, often arising when learners treat particles as separate entities. Additionally, we perform standard tokenization on this layer, such as splitting words separated by hyphens or slashes, making the tokens compatible with POS taggers.

Spelling errors Following the idea that a full system will handle spacing, punctuation, or spelling errors (e.g., Tetreault and Chodorow, 2008), we correct spelling errors, in a second tier of annotation. As with spacing errors, when spelling errors are not corrected, the correct particle cannot always be defined. Correct particles rely on correct segmentation (section 3.1), which misspellings can mask. In (3), for instance, *ki* makes it hard to determine the boundary between the stem and suffix.

(3) a. O: kalpi mas**ki**lonun
rib ???

b. C: kalpi mas-ulo-nun
rib taste-AUX-TOP
'as for rib taste'

Segmentation To know whether a particle should be used, we have to define the *position* where it could be, leading to the correct segmentation of particle-bearing words (i.e., nominals). This annotation layer builds upon the previous two: we segment corrected forms since we cannot reliably segment learner forms (cf. (3)). With segmentation, one can propose evaluating: 1) against the full correct form, or 2) against the correct particle. Note also that the important segmentation is of nominals, as we are interested in particle error detection.

3.2 Defining the target form(s)

We annotate three different categories of errors from Lee et al. (2009)—omission, replacement and addition—and one new category of errors, ordering. What we need is clarity on assigning the correct particle, i.e., the *target form*.

[2] The exact linguistic status of particles—e.g., as affixes or clitics—is a matter of some debate (see, e.g., Yoon, 2005), but is not crucial for our annotation.

[3] *Ul*/*lul*, *un*/*nun*, etc. differ phonologically.

[4] We use *O* to refer to a original form and *C* to its correction.

Defining grammaticality We follow the principle of "minimal interaction," (e.g., Hana et al., 2010): the corrected text does not have to be perfect; it is enough to be grammatical (at least for particles). One complication for defining the target particle is that particles can be dropped in spoken and even written Korean. As we focus on beginning learners who, by and large, are required to use particles, the corrected forms we annotate are obligatory within a very specific definition of *grammaticality*: they are particles which beginning learners are taught to use. Our decision captures the minimum needed for particle prediction systems and is consistent with the fact that particles are usually not dropped in formal Korean (Lee and Song, 2011).

Determining the correct particle As with English prepositions and articles, there are situations where more than one particle could be correct. In these cases, we list all reasonable alternates, allowing for a system to evaluate against a set of correct particles. There are no clear criteria for selecting one best particle out of multiple candidates, and we find low interannotator agreement in a pilot experiment, whereas we do find high agreement for a set of particles (section 4.2).

The influence of surrounding errors While many learner errors do not affect particle errors, some are relevant. For example, in (4), the verb (*uycihanta*, 'lean on') is wrong, because it requires an animate object and *sihem* ('exam') is inanimate. If we correct the verb to *tallyeissta* ('depend'), as in (4b), the correct particle is *ey*. If we do not correct the verb, the learner's particle is, in a syntactic sense, appropriate for the verb, even if the verb's selectional restrictions are not followed.

(4) a. O: nay insayng-i i sihem-**ul** uycihanta
my life-SBJ this exam-**OBJ** lean-on

b. C: nay insayng-i i sihem-**ey** tallyeissta
my life-SBJ this exam-**ON** depend

'My life depends on this exam'

It is important to clearly designate *at what point* in the process the particle is correct. Our current annotation does not deal with word choice and related semantic issues, and we thus annotate the particle at the point before any such errors are corrected. In (4), we do not correct it to (4b). Previous work has corrected sets of errors (Rozovskaya and Roth, 2010), eliminated sentences with nested or adjacent errors (Gamon, 2010), or built multiple layers of annotation (Hana et al., 2010; Boyd, 2010). Our decision makes the particle-selection task for machine learning more attainable and is easily extendible with multi-layered annotation (section 4.1).

3.3 Classifying particles

For every particle in the learner corpus, error or not, we mark its specific category, e.g., GOAL. This categorization helps because learners can make different kinds of mistakes with different kinds of particles, and systems can be developed, evaluated, or optimized with respect to a particular kind of particle.

4 Putting it together

The previous discussion outlines the type of annotation needed for evaluating Korean particle errors made by learners. As the purpose is at present to demonstrate what annotation is needed for particle error detection evaluation, we have added annotation to a small corpus. An example of full annotation is given in figure 1, for the sentence in example (5).

In the figure, positions 12 and 13 are merged to correct the spelling, as the particle (*pakkey*) was originally written as a separate token. There is a substitution error ('2' on the *Error Type* layer), with both original and correct particles noted and encoded as auxiliary particles ('A').

4.1 Annotating a corpus

We have obtained 100 learner essays from American universities, composed of 25 heritage beginners, 25 heritage intermediates, 25 foreign beginners, and 25 foreign intermediates.[5] While this is a small amount of data, it allows us to properly define the annotation scheme and show how it helps evaluation.

Table 1 provides information about the 100 essays.[6] Following previous multi-layer annotation for learner language (Lüdeling et al., 2005;

[5] The data and annotation will be available for research purposes at: `http://cl.indiana.edu/~particles/`

[6] *Raw* denotes the numbers of phenomena in the learner corpus before annotation, and *Corrected* in the fully corrected corpus., *Ecels* refer to whitespace-delimited "words".

	8	9	10	11	12	13	14	15	16	17
Token	물론	뉴욕에서	태어났기	때문에	영어	밖에	할	수	있겠죠	.
Spacing	물론	뉴욕에서	태어났기	때문에	영어밖에		할	수	있겠죠	.
Correct Spelling	물론	뉴욕에서	태어났기	때문에	영어밖에		할	수	있겠죠	.
Answer	물론	뉴욕에서	태어났기	때문에	영어만		할	수	있겠죠	.
Segmentation	물론	뉴욕+에서	태어났기	때문+에	영어+만		할	수	있겠죠	.
Error Type		0		0	2					
Original Particle		에서		에	밖에					
Correct Particle		에서		에	만					
Original Particle Type		BL		A	A					
Correct Particle Type		BL		A	A					

Figure 1: Corpus annotation for (5), using the PartiturEditor of EXMARaLDA (Schmidt, 2010)

(5) a. O: New York-eyse thayenass-ki ttaymwun-ey **yenge** **pakkey** hal swu iss-keyss-cyo.
New York-IN born-NML reason-FOR English ONLY speak be able to-FUT-END
'Since (I) was born in New York, I was able to speak only in English. '

b. C: ttaymwun-ey **yenge-man** hal ...
reason-FOR English-ONLY speak ...

Boyd, 2010), we use EXMARaLDA for encoding (Schmidt, 2010).

	Beginner		Intermediate	
	F	H	F	H
Sentences	360	376	373	297
Raw ecels	1601	2278	3483	2676
Corrected ecels	1582	2245	3392	2613
Nominals	647	949	1404	1127
Raw particles	612	808	1163	923
Corrected particles	647	887	1207	979
Omission	43	45	57	61
Substitution	60	29	47	41
Extraneous	8	8	13	5
Ordering	0	2	1	0

Table 1: Corpus Statistics (F = foreign, H = heritage)

4.2 Interannotator agreement

To gauge the reliability of the annotation, we had two experienced annotators annotate the correct particle and the error type on the heritage intermediate subcorpus, and we report the agreement on both tasks. Given the high number of times they both gave no particle to a word (in 1774 ecels), we removed these cases when calculating agreement, so as not to overly inflate the values. When either annotator used more than one particle for an instance (occurring 9 times), we only count full agreement.

The agreement rate was 94.0% for the error type (Cohen's kappa=79.1%), and 92.9% (kappa=92.3%) for specific particles. The high values can be explained by the fact that these annotators were highly-trained and were using a relatively stable set of guidelines under development for over a year (based on Lee et al. (2009)). Kappa for particle agreement is high because of the fact that there are over 30 particles, with no overwhelming majority categories, so it is unlikely for annotators to agree by chance. Previous work (Lee et al. (2009)), which did not allow multiple particles per position, had a lower agreement rate (e.g., kappa for particle value = 62%), likely due to less well-articulated guidelines.

Multiple particles To gauge how difficult it is to assign more than one particle, we selected 30 verbs that license more than two particles for a nominal argument. Using these verbs, we presented hand-constructed sentences with missing particles and asked two annotators to fill in the missing particles in the order of preference. Although the agreement rate of sets of particles was 87.8%, the agreement of the "best" particle was only 60%. This supports our decision in section 3.2 to annotate sets of particles.

References

Ghazi Abuhakema, Reem Farajand, Anna Feldman, and Eileen Fitzpatrick. 2008. Annotating an Arabic learner corpus for error. In *Proceedings of LREC 2008*. Marrakech.

Adriane Boyd. 2010. EAGLE: an error-annotated corpus of beginning learner German. In *Proceedings of LREC-10*. Malta.

Rachele De Felice and Stephen Pulman. 2008. A classifier-baed approach to preposition and determiner error correction in L2 English. In *Proceedings of COLING-08*. Manchester.

Arantza Díaz de Ilarraza, Koldo Gojenola, and Maite Oronoz. 2008. Detecting erroneous uses of complex postpositions in an agglutinative language. In *Proceedings of COLING-08*. Manchester.

Markus Dickinson, Soojeong Eom, Yunkyoung Kang, Chong Min Lee, and Rebecca Sachs. 2008. A balancing act: How can intelligent computer-generated feedback be provided in learner-to-learner interactions. *Computer Assisted Language Learning*, 21(5):369–382.

Markus Dickinson, Ross Israel, and Sun-Hee Lee. 2011. Developing methodology for Korean particle error detection. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*. Portland, OR.

Michael Gamon. 2010. Using mostly native data to correct errors in learners' writing. In *Proceedings of HLT-NAACL-10*, pages 163–171. Los Angeles, California.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2).

Jirka Hana, Alexandr Rosen, Svatava Škodová, and Barbora Štindlová. 2010. Error-tagged learner corpus of Czech. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 11–19. Uppsala, Sweden.

Hiroki Hanaoka, Hideki Mima, and Jun'ichi Tsujii. 2010. A Japanese particle corpus built by example-based annotation. In *Proceedings of LREC 2010*. Valletta, Malta.

S. Ko, M. Kim, J. Kim, S. Seo, H. Chung, and S. Han. 2004. *An analysis of Korean learner corpora and errors*. Hanguk Publishing Co.

Sun-Hee Lee, Seok Bae Jang, and Sang-Kyu Seo. 2009. Annotation of Korean learner corpora for particle error detection. *CALICO Journal*, 26(3).

Sun-Hee Lee and Jae-Young Song. 2011. Particle ellipsis in korean corpora. In *The 10th Conference for the American Association for Corpus Linguistics*. Atlanta, GA.

Anke Lüdeling, Maik Walter, Emil Kroymann, and Peter Adolphs. 2005. Multi-level error annotation in learner corpora. In *Proceedings of Corpus Linguistics 2005*. Birmingham.

Alla Rozovskaya and Dan Roth. 2010. Annotating ESL errors: Challenges and rewards. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–36. Los Angeles.

Thomas Schmidt. 2010. Linguistic tool development between community practices and technology standards. In *Proceedings of the Workshop on Language Resource and Language Technology Standards*. Malta.

Joel Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of COLING-08*. Manchester.

Joel Tetreault, Elena Filatova, and Martin Chodorow. 2010. Rethinking grammatical error annotation and evaluation with the Amazon Mechanical Turk. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*, pages 45–48. Los Angeles.

James H. Yoon. 2005. Non-morphological determination of nominal particle ordering in Korean. In L. Heggie and F. Ordonez, editors, *Clitic and Affix Combinations: Theoretical Perspectives*, pages 239–282. John Benjamins.