Building a Korean Web Corpus for Analyzing Learner Language

Markus Dickinson
Indiana University
md7@indiana.edu

Ross Israel
Indiana University
raisrael@indiana.edu

Sun-Hee Lee
Wellesley College
slee6@wellesley.edu

Abstract

Post-positional particles are a significant source of errors for learners of Korean. Following methodology that has proven effective in handling English preposition errors, we are beginning the process of building a machine learner for particle error detection in L2 Korean writing. As a first step, however, we must acquire data, and thus we present a methodology for constructing large-scale corpora of Korean from the Web, exploring the feasibility of building corpora appropriate for a given topic and grammatical construction.

1 Introduction

Applications for assisting second language learners can be extremely useful when they make learners more aware of the non-native characteristics in their writing (Amaral and Meurers, 2006). Certain constructions, such as English prepositions, are difficult to characterize by grammar rules and thus are well-suited for machine learning approaches (Tetreault and Chodorow, 2008; De Felice and Pulman, 2008). Machine learning techniques are relatively portable to new languages, but new languages bring issues in terms of defining the language learning problem and in terms of acquiring appropriate data for training a machine learner.

We focus in this paper mainly on acquiring data for training a machine learning system. In particular, we are interested in situations where the task is constant—e.g., detecting grammatical errors in particles—but the domain might fluctuate. This is the case when a learner is asked to write an essay on a prompt (e.g., “What do you hope to do in life?”), and the prompts may vary by student, by semester, by instructor, etc. By isolating a particular domain, we can hope for greater degrees of accuracy; see, for example, the high accuracies for domain-specific grammar correction in Lee and Seneff (2006).

In this situation, we face the challenge of obtaining data which is appropriate both for: a) the topic the learners are writing about, and b) the linguistic construction of interest, i.e., containing enough relevant instances. In the ideal case, one could build a corpus directly for the types of learner data to analyze. Luckily, using the web as a data source can provide such specialized corpora (Baroni and Bernardini, 2004), in addition to larger, more general corpora (Sharoff, 2006). A crucial question, though, is how one goes about designing the right web corpus for analyzing learner language (see, e.g., Sharoff, 2006, for other contexts)

The area of difficulty for language learners which we focus on is that of Korean post-positional particles, akin to English prepositions (Lee et al., 2009; Ko et al., 2004). Korean is an important language to develop NLP techniques for (see, e.g., discussion in Dickinson et al., 2008), presenting a variety of features which are less prevalent in many Western languages, such as agglutinative morphology, a rich system of case marking, and relatively free word order. Obtaining data is important in the general case, as non-English languages tend to lack resources.

The correct usage of Korean particles relies on knowing lexical, syntactic, semantic, and discourse information (Lee et al., 2005), which makes this challenging for both learners and machines (cf. English determiners in Han et al., 2006). The only other approach we know of, a parser-based one, had very low precision (Dickinson and Lee, 2009). A secondary contribution of this work is thus defining the particle error detection problem for a machine learner. It is important that the data represent the relationships between specific lexical items: in the comparable English case, for example, *interest* is usually found with *in*: ***interest*** *in/*with learning*.

The basic framework we employ is to train a machine learner on correct Korean data and then apply this system to learner text, to predict correct particle usage, which may differ from the learner's (cf. Tetreault and Chodorow, 2008). After describing the grammatical properties of particles in section 2, we turn to the general approach for obtaining relevant web data in section 3, reporting basic statistics for our corpora in section 4. We outline the machine learing set-up in section 5 and present initial results in section 6. These results help evaluate the best way to build specialized corpora for learner language.

2 Korean particles

Similar to English prepositions, Korean postpositional particles add specific meanings or grammatical functions to nominals. However, a particle cannot stand alone in Korean and needs to be attached to the preceding nominal. More importantly, particles indicate a wide range of linguistic functions, specifying grammatical functions, e.g., subject and object; semantic roles; and discourse functions. In (1), for instance, *ka* marks both the subject (function) and agent (semantic role), *eykey* the dative and beneficiary; and so forth.[1]

(1) Sumi-**ka** John-**eykey** chayk-**ul** ilhke-yo
Sumi-SBJ John-to book-OBJ read-polite
'Sumi reads a book to John.'

Particles can also combine with nominals to form modifiers, adding meanings of time, location, instrument, possession, and so forth, as shown in (2). Note in this case that the marker *ul/lul* has multiple uses.[2]

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

There are also particles associated with discourse meanings. For example, in (3) the topic marker *nun* is used to indicate old information or a discourse-salient entity, while the delimiter *to* implies that there is someone else Sumi likes. In this paper, we focus on syntactic/semantic particle usage for nominals, planning to extend to other cases in the future.

(3) Sumi-**nun** John-**to** cohahay.
Sumi-TOP John-also like
'Sumi likes John also.'

Due to these complex linguistic properties, particles are one of the most difficult topics for Korean language learners. In (4b), for instance, a learner might replace a subject particle (as in (4a)) with an object (Dickinson et al., 2008). Ko et al. (2004) report that particle errors were the second most frequent error in a study across different levels of Korean learners, and errors persist across levels (see also Lee et al., 2009).

(4) a. Sumi-*nun* chayk-*i* philyohay-yo
Sumi-TOP book-SBJ need-polite
'Sumi needs a book.'

b. *Sumi-nun chayk-**ul** philyohay-yo
Sumi-TOP book-OBJ need-polite
'Sumi needs a book.'

3 Approach

3.1 Acquiring training data

Due to the lexical relationships involved, machine learning has proven to be a good method for similar NLP problems like detecting errors in English preposition use. For example Tetreault and Chodorow (2008) use a maximum entropy classifier to build a model of correct preposition usage, with 7 million instances in their training set, and Lee and Knutsson (2008) use memory-based learning, with 10 million sentences in their training set. In expanding the paradigm to other languages, one problem

[1] We use the Yale Romanization scheme for writing Korean.

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

is a dearth of data. It seems like a large data set is essential for moving forward.

For Korean, there are at least two corpora publicly available right now, the Penn Korean Treebank (Han et al., 2002), with hundreds of thousands of words, and the Sejong Corpus (a.k.a., The Korean National Corpus, The National Institute of Korean Language, 2007), with tens of millions of words. While we plan to include the Sejong corpus in future data, there are several reasons we pursue a different tack here. First, not every language has such resources, and we want to work towards a language-independent platform of data acquisition. Secondly, these corpora may not be a good model for the kinds of topics learners write about. For example, news texts are typically written more formally than learner writing. We want to explore ways to quickly build topic-specific corpora, and Web as Corpus (WaC) technology gives us tools to do this.[3]

3.2 Web as Corpus

To build web corpora, we use BootCat (Baroni and Bernardini, 2004). The process is an iterative algorithm to bootstrap corpora, starting with various seed terms. The procedure is as follows:

1. Select initial seeds (terms).
2. Combine seeds randomly.
3. Run Google/Yahoo queries.
4. Retrieve corpus.
5. Extract new seeds via corpus comparison.
6. Repeat steps #2-#5.

For non-ASCII languages, one needs to check the encoding of webpages in order to convert the text into UTF-8 for output, as has been done for, e.g., Japanese (e.g., Erjavec et al., 2008; Baroni and Ueyama, 2004). Using a UTF-8 version of BootCat, we modified the system by using a simple Perl module (`Encode::Guess`) to look for the EUC-KR encoding of most Korean webpages and switch it to UTF-8. The pages already in UTF-8 do not need to be changed.

3.3 Obtaining data

A crucial first step in constructing a web corpus is the selection of appropriate seed terms for constructing the corpus (e.g., Sharoff, 2006; Ueyama, 2006). In our particular case, this begins the question of how one builds a corpus which models native Korean and which provides appropriate data for the task of particle error detection. The data should be genre-appropriate and contain enough instances of the particles learners know and used in ways they are expected to use them (e.g., as temporal modifiers). A large corpus will likely satisfy these criteria, but has the potential to contain distracting information. In Korean, for example, less formal writing often omits particles, thereby biasing a machine learner towards under-guessing particles. Likewise, a topic with different typical arguments than the one in question may mislead the machine. We compare the effectiveness of corpora built in different ways in training a machine learner.

3.3.1 A general corpus

To construct a general corpus, we identify words likely to be in a learner's lexicon, using a list of 50 nouns for beginning Korean students for seeds. This includes basic vocabulary entries like the words for *mother, father, cat, dog, student, teacher*, etc.

3.3.2 A focused corpus

Since we often know what domain[4] learner essays are written about, we experiment with building a more topic-appropriate corpus. Accordingly, we select a smaller set of 10 seed terms based on the range of topics covered in our test corpus (see section 6.1), shown in figure 1. As a first trial, we select terms that are, like the aforementioned general corpus seeds, level-appropriate for learners of Korean.

han-kwuk	'Korea'	*sa-lam*	'person(s)'
han-kwuk-e	'Korean (lg.)'	*chin-kwu*	'friend'
kyey-cel	'season'	*ga-jok*	'family'
hayng-pok	'happiness'	*wun-tong*	'exercise'
ye-hayng	'travel'	*mo-im*	'gathering'

Figure 1: Seed terms for the focused corpus

3.3.3 A second focused corpus

There are several issues with the quality of data we obtain from our focused terms. From an initial observation (see section 4.1), the difficulty stems in part from the simplicity of the seed terms above,

[3] Tetreault and Chodorow (2009) use the web to derive learner errors; our work, however, tries to obtain correct data.

[4] By *domain*, we refer to the subject of a discourse.

leading to, for example, actual Korean learner data. To avoid some of this noise, we use a second set of seed terms, representing relevant words in the same domains, but of a more advanced nature, i.e., topic-appropriate words that may be outside of a typical learner's lexicon. Our hypothesis is that this is more likely to lead to native, quality Korean. For each one of the simple words above, we posit two more advanced words, as given in figure 2.

kyo-sa 'teacher'	*in-kan* 'human'
phyung-ka 'evaluation'	*cik-cang* 'workplace'
pen-yuk 'translation'	*wu-ceng* 'friendship'
mwun-hak 'literature'	*sin-loy* 'trust'
ci-kwu 'earth'	*cwu-min* 'resident'
swun-hwan 'circulation'	*kwan-kye* 'relation'
myeng-sang 'meditation'	*co-cik* 'organization'
phyeng-hwa 'peace'	*sik-i-yo-pep* 'diet'
tham-hem 'exploration'	*yen-mal* 'end of a year'
cwun-pi 'preparation'	*hayng-sa* 'event'

Figure 2: Seed terms for the second focused corpus

3.4 Web corpus parameters

One can create corpora of varying size and generality, by varying the parameters given to BootCaT. We examine three parameters here.

Number of seeds The first way to vary the type and size of corpus obtained is by varying the number of seed terms. The exact words given to BootCaT affect the domain of the resulting corpus, and utilizintg a larger set of seeds leads to more potential to create a bigger corpus. With 50 seed terms, for example, there are 19,600 possible 3-tuples, while there are only 120 possible 3-tuples for 10 seed terms, limiting the relevant pages that can be returned.

For the general (G) corpus, we use: G1) all 50 seed terms, G2) 5 sets of 10 seeds, the result of splitting the 50 seeds randomly into 5 buckets, and G3) 5 sets of 20 seeds, which expand the 10-seed sets in G2 by randomly selecting 10 other terms from the remaining 40 seeds. This breakdown into 11 sets (1 G1, 5 G2, 5 G3) allows us to examine the effect of using different amounts of general terms and facilitates easy comparison with the first focused corpus, which has only 10 seed terms.

For the first focused (F_1) corpus, we use: $F_1$1) the 10 seed terms, and $F_1$2) 5 sets of 20 seeds, obtained by combining $F_1$1 with each seed set from G2. This second group provides an opportunity to examine what happens when augmenting the focused seeds with more general terms; as such, this is a first step towards larger corpora which retain some focus. For the second focused corpus (F_2), we simply use the set of 20 seeds. We have 7 sets here (1 $F_1$1, 5 $F_1$2, 1 F_2), giving us a total of 18 seed term sets at this step.

Tuple length One can also experiment with tuple length in BootCat. The shorter the tuple, the more webpages that can potentially be returned, as short tuples are likely to occur in several pages (e.g., compare the number of pages that all of *person happiness season* occur in vs. *person happiness season exercise travel*). On the other hand, longer tuples are more likely truly relevant to the type of data of interest, more likely to lead to well-formed language. We experiment with tuples of different lengths, namely 3 and 5. With 2 different tuple lengths and 18 seed sets, we now have 36 sets.

Number of queries We still need to specify how many queries to send to the search engine. The maximum number is determined by the number of seeds and the tuple size. For 3-word tuples with 10 seed terms, for instance, there are 10 items to choose 3 objects from: $\binom{10}{3} = \frac{10!}{3!(10-3)!} = 120$ possibilities.

Using all combinations is feasible for small seed sets, but becomes infeasible for larger seed sets, e.g., $\binom{50}{5} = 2,118,760$ possibilities. To reduce this, we opt for the following: for 3-word tuples, we generate 120 queries for all cases and 240 queries for the conditions with 20 and 50 seeds. Similarly, for 5-word tuples, we generate the maximum 252 queries with 10 seeds, and both 252 and 504 for the other conditions. With the previous 36 sets (12 of which have 10 seed terms), evenly split between 3 and 5-word tuples, we now have 60 total corpora, as in table 1.

		# of seeds					
tuple	# of	General			F_1		F_2
len.	queries	10	20	50	10	20	20
3	120	5	5	1	1	5	1
	240	n/a	5	1	n/a	5	1
5	252	5	5	1	1	5	1
	504	n/a	5	1	n/a	5	1

Table 1: Number of corpora based on parameters

Other possibilities There are other ways to increase the size of a web corpus using BootCaT. First, one can increase the number of returned pages for a particular query. We set the limit at 20, as anything higher will more likely result in non-relevant data for the focused corpora and/or duplicate documents.

Secondly, one can perform iterations of searching, extracting new seed terms with every iteration. Again, the concern is that by iterating away from the initial seeds, a corpus could begin to lose focus. We are considering both extensions for the future.

Language check One other constraint we use is to specify the particular language of interest, namely that we want Korean pages. This parameter is set using the language option when collecting URLs. We note that a fair amount of English, Chinese, and Japanese appears in these pages, and we are currently developing our own Korean filter.

4 Corpus statistics

To gauge the properties of size, genre, and degree of particle usage in the corpora, independent of application, basic statistics of the different web corpora are given in table 2, where we average over multiple corpora for conditions with 5 corpora.[5]

There are a few points to understand in the table. First, it is hard to count true words in Korean, as compounds are frequent, and particles have a debatable status. From a theory-neutral perspective, we count *ejel*s, which are tokens occurring between white spaces. Secondly, we need to know about the number of particles and number of nominals, i.e., words which could potentially bear particles, as our machine learning paradigm considers any nominal a test case for possible particle attachment. We use a POS tagger (Han and Palmer, 2004) for this.

Some significant trends emerge when comparing the corpora in the table. First of all, longer queries (length 5) result in not only more returned unique webpages, but also longer webpages on average than shorter queries (length 3). This effect is most dramatic for the F_2 corpora. The F_2 corpora also exhibit a higher ratio of particles to nominals than the other web corpora, which means there will be more positive examples in the training data for the machine learner based on the F_2 corpora.

4.1 Qualitative evaluation

In tandem with the basic statistics, it is also important to gauge the quality of the Korean data from a more qualitative perspective. Thus, we examined the 120 3-tuple F_1 corpus and discovered a number of problems with the data.

First, there are issues concerning collecting data which is not pure Korean. We find data extracted from Chinese travel sites, where there is a mixture of non-standard foreign words and unnatural-sounding translated words in Korean. Ironically, we also find learner data of Korean in our search for correct Korean data. Secondly, there are topics which, while exhibiting valid forms of Korean, are too far afield from what we expect learners to know, including religious sites with rare expressions; poems, which commonly drop particles; gambling sites; and so forth. Finally, there are cases of ungrammatical uses of Korean, which are used in specific contexts not appropriate for our purposes. These include newspaper titles, lists of personal names and addresses, and incomplete phrases from advertisements and chats. In these cases, we tend to find less particles.

Based on these properties, we developed the aforementioned second focused corpus with more advanced Korean words and examined the 240 3-tuple F_2 corpus. The F_2 seeds allow us to capture a greater percentage of well-formed data, namely data from news articles, encyclopedic texts, and blogs about more serious topics such as politics, literature, and economics. While some of this data might be above learners' heads, it is, for the most part, well-formed native-like Korean. Also, the inclusion of learner data has been dramatically reduced. However, some of the same problems from the F_1 corpus persist, namely the inclusion of poetry, newspaper titles, religious text, and non-Korean data.

Based on this qualitative analysis, it is clear that we need to filter out more data than is currently being filtered, in order to obtain valid Korean of a type which uses a sufficient number of particles in grammatical ways. In the future, we plan on restricting the genre, filtering based on the number of rare words (e.g., religious words), and using a trigram language model to check the validity.

[5] For the 252 5-tuple 20 seed General corpora, we average over four corpora, due to POS tagging failure on the fifth corpus.

					Ejel		Particles		Nominals	
Corpus	Seeds	Len.	Queries	URLs	Total	Avg.	Total	Avg.	Total	Avg.
Gen.	10	3	120	1096.2	1,140,394.6	1044.8	363,145.6	331.5	915,025	838.7
		5	252	1388.2	2,430,346.4	1779.9	839,005.8	618.9	1,929,266.0	1415.3
	20	3	120	1375.2	1,671,549.2	1222.1	540,918	394.9	1,350,976.6	988.6
		3	240	2492.4	2,735,201.6	1099.4	889,089	357.3	2,195,703	882.4
		5	252	1989.6	4,533,642.4	2356	1,359,137.2	724.5	3,180,560.6	1701.5
		5	504	3487	7,463,776	2193.5	2,515,235.8	741.6	5,795,455.8	1709.7
	50	3	120	1533	1,720,261	1122.1	584,065	380.9	1,339,308	873.6
		3	240	2868	3,170,043	1105.3	1,049,975	366.1	2,506,995	874.1
		5	252	1899.5	4,380,684.2	2397.6	1,501,358.7	821.5	3,523,746.2	1934.6
		5	504	5636	5,735,859	1017.7	1,773,596	314.6	4,448,815	789.3
F_1	10	3	120	1315	628,819	478.1	172,415	131.1	510,620	388.3
		5	252	1577	1,364,885	865.4	436,985	277.1	1,069,898	678.4
	20	3	120	1462.6	1,093,772.4	747.7	331,457.8	226.8	885,157.2	604.9
			240	2637.2	1,962,741.8	745.2	595,570.6	226.1	1,585,730.4	602.1
		5	252	2757.6	2,015,077.8	730.8	616,163.8	223.4	1,621,306.2	588
			504	4734	3,093,140.4	652.9	754,610	159.8	1,993,104.4	422.1
F_2	20	3	120	1417	1,054,925	744.5	358,297	252.9	829,416	585.3
			240	2769	1,898,383	685.6	655,757	236.8	1,469,623	530.7
		5	252	1727	4,510,742	2611.9	1,348,240	780.7	2,790,667	1615.9
			504	2680	6,916,574	2580.8	2,077,171	775.1	4,380,571	1634.5

Table 2: Basic statistics of different web corpora

Note that one might consider building even larger corpora from the start and using the filtering step to winnow down the corpus for a particular application, such as particle error detection. However, while removing ungrammatical Korean is a process of removing noise, identifying whether a corpus is about traveling, for example, is a content-based decision. Given that this is what a search engine is designed to do, we prefer filtering based only on grammatical and genre properties.

5 Classification

We describe the classification paradigm used to determine how effective each corpus is for detecting correct particle usage; evaluation is in section 6.

5.1 Machine learning paradigm

Based on the parallel between Korean particles and English prepositions, we use preposition error detection as a starting point for developing a classifier. For prepositions, Tetreault and Chodorow (2008) extract 25 features to guess the correct preposition (out of 34 selected prepositions), including features capturing the lexical and grammatical context (e.g., the words and POS tags in a two-word window around the preposition) and features capturing various relevant selectional properties (e.g., the head verb and noun of the preceding VP and NP).

We are currently using TiMBL (Daelemans et al., 2007) for development purposes, as it provides a range of options for testing. Given that learner data needs to be processed instantaneously and that memory-based learning can take a long time to classify, we will revisit this choice in the future.

5.2 Defining features

5.2.1 Relevant properties of Korean

As discussed in section 2, Korean has major differences from English, leading to different features. First, the base word order of Korean is SOV, which means that the following verb and following noun could determine how the current word functions. However, since Korean allows for freer word order than English, we do not want to completely disregard the previous noun or verb, either.

Secondly, the composition of words is different than English. Words contain a stem and an arbitrary number of suffixes, which may be derivational morphemes as well as particles, meaning that we must consider sub-word features, i.e., segment words into their component morphemes.

Finally, particles have more functions than prepositions, requiring a potentially richer space of features. Case marking, for example, is even more dependent upon the word's grammatical function in a sentence. In order to ensure that our system can correctly handle all of the typical relations between words without failing on less frequent constructions, we need (large amounts of) appropriate data.

5.2.2 Feature set

To begin with, we segment and POS tag the text, using a hybrid (trigram + rule-based) morphological tagger for Korean (Han and Palmer, 2004). This segmentation phase means that we can define subword features and isolate the particles in question. For our features, we break each word into: a) its stem and b) its combined affixes (excluding particles), and each of these components has its own POS, possibly a combined tag (e.g., EPF+EFN), with tags from the Penn Korean Treebank (Han et al., 2002).

The feature vector uses a five word window that includes the target word and two words on either side for context. Each word is broken down into four features: stem, affixes, stem_POS, and affixes_POS. Given the importance of surrounding noun and verbs for attachment in Korean, we have features for the preceding as well as the following noun and verb. For the noun/verb features, only the stem is used, as this is largely a semantically-based property.

In terms of defining a class, if the target word's affixes contain a particle, it is removed and used as the basis for the class; otherwise the class is NONE. We also remove particles in the context affixes, as we cannot rely on surrounding learner particles.

As an example, consider predicting the particle for the word *Yenge* ('English') in (5a). We generate the instance in (5b). The first five lines refer to the previous two words, the target word, and the following two words, each split into stem and suffixes along with their POS tags, and with particles removed. The sixth line contains the stems of the preceding and following noun and verb, and finally, there is the class (YES/NO).

(5) a. Mikwuk-*eyse* sal-*myense* Yenge-*man-ul* cip-*eyse* ss-*ess-eyo*.
America-in live-while English-only-OBJ home-at use-Past-Decl
'While living in America, (I/she/he) used only English at home.'

b. Mikwuk NPR NONE NONE
sal VV myense ECS
Yenge NPR NONE NONE
cip NNC NONE NONE
ss VV ess+eyo EPF+EFN
sal Mikwuk ss cip
YES

For the purposes of evaluating the different corpora, we keep the task simple and only guess YES or NO for the existence of a particle. We envision this as a first pass, where the specific particle can be guessed later. This is also a practical task, in that learners can benefit from accurate feedback on knowing whether or not a particle is needed.

6 Evaluation

We evaluate the web corpora for the task of predicting particle usage, after describing the test corpus.

6.1 Learner Corpus

To evaluate, we use a corpus of learner Korean made up of essays from college students (Lee et al., 2009). The corpus is divided according to student level (beginner, intermediate) and student background (heritage, non-heritage),[6] and is hand-annotated for particle errors. We expect beginners to be less accurate than intermediates and non-heritage less accurate than heritage learners. To pick a middle ground, the current research has been conducted on non-heritage intermediate learners. The test corpus covers a range of common language classroom topics such as Korean language, Korea, friends, family, and traveling.

We run our system on raw learner data, i.e, unsegmented and with spelling and spacing errors included. As mentioned in section 5.2.2, we use a POS tagger to segment the words into morphemes, a crucial step for particle error detection.[7]

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

[7] In the case of segmentation errors, we cannot possibly get the particle correct. We are currently investigating this issue.

	Seeds	Len.	Quer.	P	R	F
Gen.	10	3	120	81.54%	76.21%	78.77%
		5	252	82.98%	77.77%	80.28%
	20	3	120	81.56%	77.26%	79.33%
		3	240	82.89%	78.37%	80.55%
		5	252	83.79%	78.17%	80.87%
		5	504	84.30%	79.44%	81.79%
	50	3	120	82.97%	77.97%	80.39%
		3	240	83.62%	80.46%	82.00%
		5	252	82.57%	78.45%	80.44%
		5	504	84.25%	78.69%	81.36%
F_1	10	3	120	81.41%	74.67%	77.88%
		5	252	83.82%	77.09%	80.30%
	20	3	120	82.23%	76.40%	79.20%
			240	82.57%	77.19%	79.78%
		5	252	83.62%	77.97%	80.68%
			504	81.86%	75.88%	78.73%
F_2	20	3	120	81.63%	76.44%	78.93%
			240	82.57%	78.45%	80.44%
		5	252	84.21%	80.62%	82.37%
			504	83.87%	81.51%	82.67%

Table 3: Results of guessing particle existence, training with different corpora

The non-heritage intermediate (NHI) corpus gives us 3198 words, with 1288 particles and 1836 nominals. That is, about 70% of the nominals in the learner corpus are followed by a particle. This is a much higher average than in the 252 5-tuple F_2 corpus, which exhibits the highest average of all of the web corpora at about 48% ($\frac{781}{1616}$; see table 2).

6.2 Results

We use the default settings for TiMBL for all the results we report here. Though we have obtained 4-5% higher F-scores using different settings, the comparisons between corpora are the important measure for the current task. The results are given in table 3.

The best results were achieved when training on the 5-tuple F_2 corpora, leading to F-scores of 82.37% and 82.67% for the 252 tuple and 504 tuple corpora, respectively. This finding reinforces our hypothesis that more advanced seed terms result in more reliable Korean data, while staying within the domain of the test corpus. Both longer tuple lengths and greater amounts of queries have an effect on the reliability of the resulting corpora. Specificaly, 5-tuple corpora produce better results than similar 3-tuple corpora, and corpora with double the amount of queries of n-length perform better than smaller comparable corpora. Although larger corpora tend to do better, it is important to note that there is not a clear relationship. The general 50/5/252 corpus, for instance, is similarly-sized to the F_2 focused 20/5/252 corpus, with over 4 million ejels (see table 2). The focused corpus—based on fewer yet more relevant seed terms—has 2% better F-score.

7 Summary and Outlook

In this paper, we have examined different ways to build web corpora for analyzing learner language to support the detection of errors in Korean particles. This type of investigation is most useful for lesser-resourced languages, where the error detection task stays constant, but the topic changes frequently. In order to develop a framework for testing web corpora, we have also begun developing a machine learning system for detecting particle errors.

The current web data, as we have demonstrated, is not perfect, and thus we need to continue improving that. One approach will be to filter out clearly non-Korean data, as suggested in section 4.1. We may also explore instance sampling (e.g., Wunsch et al., 2009) to remove many of the non-particle nominal (negative) instances, which will reduce the difference between the ratios of negative-to-positive instances of the web and learner corpora. We still feel that there is room for improvement in our seed term selection, and plan on constructing specific web corpora for each topic covered in the learner corpus. We will also consider adding currently available corpora, such as the Sejong Corpus (The National Institute of Korean Language, 2007), to our web data.

With better data, we can work on improving the machine learning system. This includes optimizing the set of features, the parameter settings, and the choice of machine learning algorithm. Once the system has been optimized, we will need to test the results on a wider range of learner data.

Acknowledgments

We would like to thank Marco Baroni and Jan Pomikálek for kindly providing a UTF-8 version of BootCat; Chong Min Lee for help with the POS tagger, provided by Chung-Hye Han; and Joel Tetreault for useful discussion.

References

Amaral, Luiz and Detmar Meurers (2006). Where does ICALL Fit into Foreign Language Teaching? Talk given at CALICO Conference. May 19, 2006. University of Hawaii.

Baroni, Marco and Silvia Bernardini (2004). BootCaT: Bootstrapping Corpora and Terms from the Web. In *Proceedings of LREC 2004*. pp. 1313–1316.

Baroni, Marco and Motoko Ueyama (2004). Retrieving Japanese specialized terms and corpora from the World Wide Web. In *Proceedings of KONVENS 2004*.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, Antal van den Bosch, Timbl Tilburg and Memory based Learner (2007). TiMBL: Tilburg Memory-Based Learner - version 6.1 - Reference Guide.

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

Dickinson, Markus, 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.

Dickinson, Markus and Chong Min Lee (2009). Modifying Corpus Annotation to Support the Analysis of Learner Language. *CALICO Journal* 26(3).

Erjavec, Irena Srdanoviĉ, Tomaz Erjavec and Adam Kilgarriff (2008). A Web Corpus and Word Sketches for Japanese. *Information and Media Technologies* 3(3), 529–551.

Han, Chung-Hye, Na-Rare Han, Eon-Suk Ko and Martha Palmer (2002). Development and Evaluation of a Korean Treebank and its Application to NLP. In *Proceedings of LREC-02*.

Han, Chung-Hye and Martha Palmer (2004). A Morphological Tagger for Korean: Statistical Tagging Combined with Corpus-Based Morphological Rule Application. *Machine Translation* 18(4), 275–297.

Han, Na-Rae, Martin Chodorow and Claudia Leacock (2006). Detecting Errors in English Article Usage by Non-Native Speakers. *Natural Language Engineering* 12(2).

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

Lee, John and Ola Knutsson (2008). The Role of PP Attachment in Preposition Generation. In *Proceedings of CICLing 2008*. Haifa, Israel.

Lee, John and Stephanie Seneff (2006). Automatic Grammar Correction for Second-Language Learners. In *INTERSPEECH 2006*. Pittsburgh, pp. 1978–1981.

Lee, Sun-Hee, Donna K. Byron and Seok Bae Jang (2005). Why is Zero Marking Important in Korean? In *Proceedings of IJCNLP-05*. Jeju Island, Korea.

Lee, Sun-Hee, Seok Bae Jang and Sang kyu Seo (2009). Annotation of Korean Learner Corpora for Particle Error Detection. *CALICO Journal* 26(3).

Sharoff, Serge (2006). Creating General-Purpose Corpora Using Automated Search Engine Queries. In *WaCky! Working papers on the Web as Corpus. Gedit*.

Tetreault, Joel and Martin Chodorow (2008). The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proceedings of COLING-08*. Manchester.

Tetreault, Joel and Martin Chodorow (2009). Examining the Use of Region Web Counts for ESL Error Detection. In *Web as Corpus Workshop (WAC-5)*. San Sebastian, Spain.

The National Institute of Korean Language (2007). The Sejong Corpus.

Ueyama, Motoko (2006). Evaluation of Japanese Web-based Reference Corpora: Effects of Seed Selection and Time Interval. In *WaCky! Working papers on the Web as Corpus. Gedit*.

Wunsch, Holger, Sandra Kübler and Rachael Cantrell (2009). Instance Sampling Methods for Pronoun Resolution. In *Proceedings of RANLP 2009*. Borovets, Bulgaria.