The development of a web corpus of Hindi language and corpus-based comparative studies to Japanese

Miki Nishioka
Osaka University
Osaka
Japan
dumas@lang.osaka-u.ac.jp

Shiro Akasegawa
Lago Language Institute
Shiga
Japan
lagoinst@gmail.com

Abstract

In this paper, we discuss our creation of a web corpus of spoken Hindi (COSH), one of the Indo-Aryan languages spoken mainly in the Indian subcontinent. We also point out notable problems we've encountered in the web corpus and the special concordancer. After observing the kind of technical problems we encountered, especially regarding annotation tagged by Shiva Reddy's tagger, we argue how they can be solved when using COSH for linguistic studies. Finally, we mention the kinds of linguistic research that we non-native speakers of Hindi can do using the corpus, especially in pragmatics and semantics, and from a comparative viewpoint to Japanese.

1 Introduction

Hindi-Urdu is a member of the Indo-Aryan language family widely distributed in Indian subcontinent. It is originally related the Indo-Iranian branch of the Indo-European language family. In contrast, Japanese is an East Asian language spoken mainly in Japan. Genealogically, geographically, and even historically, Japanese has no direct relation to Hindi-Urdu, with Japanese lacking the declension of nouns, pronouns, adjectives and verbs based on person, gender and number peculiar to the languages such as Hindi-Urdu. Nevertheless, morpho-syntactically and semantically, both languages have many common features, such as the word order of a simple sentence and a compound sentence (except a complex sentence). Other common features are: complex predicates (basically 'noun + light verb' as in H. *paRhaaii karnaa* vs. J. *benkyou suru* both for 'studying do'), noun modification with participles of a verb, nominalization with genitive particle *no* in Japanese, and *vaalaa* and genitive postposition *kaa* in Hindi, and verb-verb concatenation, that is, so-called 'compound verbs' (CV). These features are similar (analogical), but not homological: alike in appearance, and working similarly but not exactly the same. It is sometimes hard for us non-native speakers of Hindi-Urdu (hereafter simply 'Hindi') to understand nuances of meaning, since we lack intuition of the language.

To make up for lack of intuition, large-scale corpora will make useful tools for language study. In this paper, we will discuss how we created a web corpus of the Hindi language, and the kind of concordancer that we have developed. Based on these, we will give a few examples of comparative studies of Hindi and Japanese.

2 General methods for studying a foreign language; their pros and cons

Before introducing the corpus and the concordancer, we should discuss general methods for studying a foreign language. When studying a foreign language, meaning a non-native language, there are three basic methods for investigating linguistic phenomena:

a. Finding a handful of native speakers of the target language and interviewing them about a linguistic topic.

Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing,
pages 114–123, Osaka, Japan, December 11-17 2016.

b. Conducting a questionnaire about the topic and collecting results from a larger population.
c. Using a large corpus and collecting results from a much larger population automatically. This last method is useful for proving a linguistic phenomenon objectively.

Method a. is an orthodox and common way for linguists to study a foreign language. It is quite convenient, as finding a nearby native speaker of the target language is easy, especially nowadays. However, the number of results is too small to be scientifically valid. Method b. is used to make up for this lack of objectivity. It too is an orthodox way of investigating a target language. However, the number of results it yields is still small, and the results themselves tend to fluctuate depending on the pre-set questions and answers.

How about method c: using a large corpus? This method lets us easily collect results from a far larger population. It is one of the more promising ways to qualify results, eliminate subjectivity as much as possible for investigative purposes, and provide objectivity and reliability. However, it's not without its problems. Undeniably, a corpus still has some imbalance in genres of the language. In addition, it’s difficult to build a large corpus in a short time. Furthermore, technical problems can arise, especially when tagging and annotating words in the corpus.

To maintain or increase the quality of results, we should bear in mind that each of the methods in itself is insufficient, since all have both strong and weak points. For example, method a. can produce imbalances in data based on the researcher’s own personal experience of the language, unless the informant is well-trained in both the language and linguistic research. This kind of useful informant is rather rare. Thus, utilizing all three methods effectively is the key for future research, especially on non-native languages.

3 Development of a web-corpus and a concordancer

Since I keenly felt the need for a large-scale corpus for semantic or pragmatic research on Hindi, I have devoted considerable effort to building a web corpus, with the kind technical assistance of Shiro Akasegawa. A development version of this corpus will be open to the public in October, 2016. The present size of the corpus is 179,979,464 tokens. In preparing to build the corpus, we encountered several technical issues in dealing with Devanagari script online. These are described below.

3.1 Pre-treatment for building the Hindi web corpus

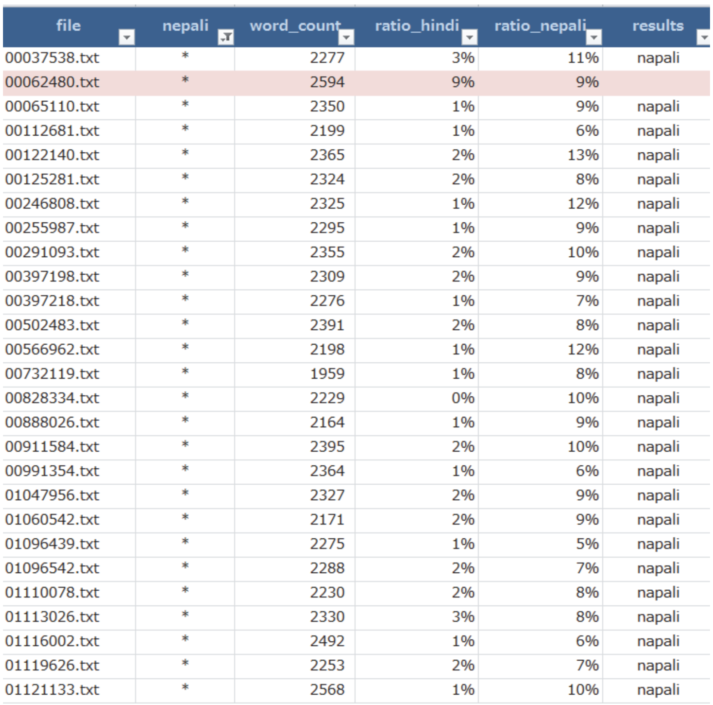
In the process of preparing the Hindi web corpus, we faced several problems. As we know, Hindi is not the only language to use Devanagari script: there are Nepali, Marathi, Maithili, Rajasthani, Bhojpuri, Sanskrit, and others. Thus, in our collecting of data in Devanagari script, some Nepali, Sanskrit, and Maithili data slipped in with them. Sanskrit is easy to tell apart, as its words tend to combine according to Sandhi rules, emphasizing a definite phonetic harmony. Other languages, however, look the same as Hindi, with a blank space to divide words. To weed these out, we took steps as below.

Judging whether a language was Hindi or not was done as follows:

a. We targeted Hindi, Nepali, Rajsthani, and Bhojpuri
b. We made a lexical list for each language, choosing 100 frequently-used lexicons. The list calculates the frequency of the lexicons as contained in corpora texts.
c. Using a random sampling technique, we prepared 637 text files of Bhojpuri, Maithili, Marathi, Nepali, Rajsthani, and Sanskrit, all of which are generally written in Devanagari script.

We checked whether the languages of the texts and each high-scoring language were the same. We found that language identification failed only in one Nepali case. The number of Nepali files was 27, of which 26 had a higher rate of Hindi lexicons than of Nepali lexicons. Only one file had the same rate of Hindi and Nepali lexicons (9%).

The determining criterion: analyzing these results, we found that all files with over **20%** frequency of Hindi lexicons were in Hindi. Therefore, we decided the criterion was valid, and used it to identify languages. Only files found to be acceptable based on this criterion were included in the eventual corpus.



file	nepali	word_count	ratio_hindi	ratio_nepali	results
00037538.txt	*	2277	3%	11%	napali
00062480.txt	*	2594	9%	9%	
00065110.txt	*	2350	1%	9%	napali
00112681.txt	*	2199	1%	6%	napali
00122140.txt	*	2365	2%	13%	napali
00125281.txt	*	2324	2%	8%	napali
00246808.txt	*	2325	1%	12%	napali
00255987.txt	*	2295	1%	9%	napali
00291093.txt	*	2355	2%	10%	napali
00397198.txt	*	2309	2%	9%	napali
00397218.txt	*	2276	1%	7%	napali
00502483.txt	*	2391	2%	8%	napali
00566962.txt	*	2198	1%	12%	napali
00732119.txt	*	1959	1%	8%	napali
00828334.txt	*	2229	0%	10%	napali
00888026.txt	*	2164	1%	9%	napali
00911584.txt	*	2395	2%	10%	napali
00991354.txt	*	2364	1%	6%	napali
01047956.txt	*	2327	2%	9%	napali
01060542.txt	*	2171	2%	9%	napali
01096439.txt	*	2275	1%	5%	napali
01096542.txt	*	2288	2%	7%	napali
01110078.txt	*	2230	2%	8%	napali
01113026.txt	*	2330	3%	8%	napali
01116002.txt	*	2492	1%	6%	napali
01119626.txt	*	2253	2%	7%	napali
01121133.txt	*	2568	1%	10%	napali

Table 1: A sample of the determining criterion

Another prominent problem we had to face was **duplicated data** found in the first raw web corpus. We divided the corpus files into sentence units, sorted them, and deleted the duplicates. From an initial 12,170,339 sentences, we ended up with 8,806,658 sentences, meaning 3,360,000 duplicates, or about 28% of all sentences.

Another problem is that we left out Hindi data in the Roman alphabet. The Internet features copious Hindi data in Roman alphabet, providing precious linguistic material in natural Hindi (which might also be called 'Urdu'). However, Hindi Romanization is vastly inconsistent, and since we found few if any established rules, we decided to exclude those data for the time being.

3.2 Annotation by a POS tagger

To annotate the Hindi data in our web corpus, we chose Shiva Reddy's POS tagger[1] implemented for Sketch Engine[2]. According to Reddy, this tagger achieves 91.31% accuracy, trained on a corpus of 30,409,730 tokens[3]. However, our web corpus consists of natural language, and tends to contain numerous new loanwords from other languages, written in Devanagari – which the tagger cannot tag properly. Moreover, typographic errors are commonplace on the Internet, because no fixed orthography has taken root among common people, unlike in Japanese. These errors should prevent the tagger from achieving 91.31% accuracy on tokens in our web corpus. The real accuracy in the web corpus should be lower.

Another big problem is Hindi itself. There are many homographs in Hindi. As Dalal et al (2007) have mentioned, these are longstanding problems in computational linguistics. There are some patterns of ambiguity. Some prominent ones are mentioned below.

Ambiguity of categories (POS)

[1] Available at http://sivareddy.in/downloads

[2] Hosted at https://www.sketchengine.co.uk/

[3] See Hindi Part of Speech (POS) Tagger, at https://bitbucket.org/sivareddyg/hindi-part-of-speech-tagger, accessed Aug 7, 2016.

A notable example of this type is the homograph *aam*. It has two meanings: an adjective [JJ] 'general' and a masculine noun [NN] 'mango'. However, the word order of a noun phrase is fixed: an adjective comes before a noun in Hindi. So it's easy to tell which is JJ and which is NN, especially on the basis of the trigram and the probability of the POS attached to the tagger. Thus, this example poses a rather minor problem.

Ambiguity of forms in the same POS category

What the tagger cannot distinguish easily is a homograph with various forms in the same POS category. One example is the verb *baiTh-naa*. The annotated part cited in Table 2 is *udaas baiTh-aa hai*. 'He is sitting sadly.'

surface form	lemma	tag	details[4]	POS	gender	number	person	case
उदास	उदास	NN	----	adj	any	any	any	any
बैठा	बैठा	VM	0	v	any	any	any	--
है	है	VAUX	है	v	any	sg	2	--

Table 2: Example of annotation by Shiva Reddy's tagger

Putting aside *udaas,* which here is annotated NN, *baiTh-aa*, the perfect participle form here consisting of a stem and a perfect participle or past [for both masculine and singular] suffix *–aa,* is annotated as VM [0], meaning a stem form – i.e., the form with no suffixes. In other words, it indicates a stem form of *baiTh**aa**-naa*[5] or *biTh**aa**-naa* 'to make someone sit', even though it should be the perfect participle *baiTh-aa*, which is derived from the verb intransitive *baiTh-naa*. This is true of such verb pairs as *ban-naa* 'to be made' (intransitive) vs. *ban**aa**-naa* 'to make' (transitive), and *cal-naa* 'to move' (intransitive) vs. *cal**aa**-naa* 'to move '(transitive). We have also found examples such as *samajh-naa* 'to understand' vs. *samjh**aa**-naa* 'to cause to understand', *sun-naa* 'to hear' vs. *sun**aa**-naa* 'to cause to hear', and *pahan-naa* 'to wear, to put on' vs. *pahan**aa**-naa* 'to cause to put on'.

Regarding homographs, we can provide a couple of more examples such as verbal nouns and a finite form of the same verb. The verb *khaa-naa* 'to eat' in the infinitive is identical to the verbal noun or noun *khaanaa* 'food' itself. However, in all of our randomly chosen samples, the tagger has distinguished the noun form from the infinitive form successfully. In addition, there are other representative verbal nouns with an *-ii* ending, such as *paRhaa-ii* 'studying', *sunaa-ii* 'hearing', and *dikhaa-ii* 'seeing'. However, *paRhaa-ii* is annotated as successfully as *khaanaa* 'food' above. The latter two forms are used in Noun + *de-naa* 'to give'/ *paR-naa* 'to fall'; that is, in so-called complex predicates. This might be the reason why the verbal nouns are annotated as VM, not NN.

Some complex tagging and POS details

In addition to the above, what we must point out is the complex tagging by the tagger. For example, regarding so-called complex predicates, the slots consist of Slot 1 + Slot 2. Options for slot 1 are Noun, Adjective and Verb. Slot 2 is mostly filled with so-called light verbs. There are some primary light verbs, that is V2, such as *kar-naa* 'to do', *ho-naa* 'to be', *le-naa* 'to take', *de-naa* 'to give', etc. Of these V2s, when it comes to *de-naa* 'to give', we find that certain nouns such as *dhokhaa* 'deceit' and *udhaar* 'a loan, debt', as in *dhokhaa de-naa* 'to deceive' and *udhaar de-naa* 'to lend', respectively, are labelled as VM; while verbs for Slot 2 are VAUX.

Another thing to mention is that adjectives with *-aa* endings have separate masculine and feminine forms. Following, from top to bottom, is an example of the adjective बड़ा *baRaa* 'big' in the masculine (m) and singular (sg) form, which is the default or lemma form[6]; बड़े *baRe* in the masculine and singular plus the oblique case (o); and बड़ी *baRii* in the feminine (f) and either singular or plural, i.e., (any).

[4] Here suffixes like the infinitive –ना (*-naa*), imperfect participle –ता (*-taa*), and perfect participle -या (*-yaa*) are added in this column optionally.

[5] The suffix ***aa*** in the verb is added to make an intransitive verb into a transitive. The same applies hereafter.

[6] A lemma form, that is, an unmarked word in Hindi, is treated as a direct case (d), not in the oblique case (o).

surface form	lemma	tag	details	POS	gender	number	person	case
बड़ा	बडा	JJ	----	adj	m	sg	--	d
बड़े	बडा	JJ	----	adj	m	sg	--	o
बड़ी	बडी	JJ	----	adj	f	any	--	any

Table 3: A sample of annotation for adjective *baRaa* ‘big’

As we see in Table 3, *baRe* in the second line is identical to *baRe* the masculine and plural form. However, the tagger tends to annotate *baRe* as the same form in the oblique case. In addition, lemma forms here are without ◌़ (*nuqtaa*), the dot under each character: बडा *baDaa* and बडी *baDii*. In any event, the lemma of *baDii* should basically be the same as that of *baDaa,* and yet the tagger keeps the feminine form for *baDii*. We have another example of *baRaa*, *baRe* and *baRii,* as shown in Table 4 below.

surface form	lemma	tag	details	POS	gender	number	person	case
बड़ा	बडा	XC	----	punc	--	--	--	--
बड़े	बडे	NN	----	punc	--	--	--	--
बड़ी	बडी	XC	----	punc	--	--	--	--

Table 4: Another sample of annotation for Adjective *baRaa* ‘big’

The surface form and the lemma form in the first and third lines are the same as in Table 3. However, the word is tagged as (XC), that is, compound[7]. The second *baRe* is annotated as a noun (NN), though the POS is labeled as a punctuation. The lemma form is also *baDe*, which is different from the pattern of Table 3. A similar example of adjectives is अच्छा *acchaa* ‘good’.

surface form	lemma	tag	details	POS	gender	number	person	case
अच्छा	अच्छा	JJ	----	adj	m	sg	--	d
अच्छे	अच्छे	JJ	----	adj	any	any	--	any
अच्छी	अच्छी	JJ	----	adj	f	any	--	any

Table 5: A sample of annotation for Adjective *acchaa* ‘good’

Acchaa has three different lemma forms in the first place, अच्छा *acchaa*, अच्छे *acche*, अच्छी *acchii*, for (m) + (sg), (m) + (any), and (f) + (any), respectively. Although all lemma forms of *-aa* adjectives should be only *-aa* forms, they are tagged like this.

From what we’ve seen here, it is necessary for users to understand these facts, such as ambiguities and tagging problems, when searching certain words by lemma or by tag.

3.3 Pre-treatment for developing a concordancer

In order to release the web corpus, we planned to develop a special concordancer. Before developing it, we tried running a search using a Perl script, and found additional technical problems requiring attention. Of these, Unicode Devanagari character processing and its character codes was the most difficult to solve. There are two ways to type a character with a *nuqtaa* dot. We can type ड़ in two ways: as 0921 (ड) + 093C (◌़), i.e., *nuqtaa;* and as 095C (ड़). The problem is that when the tagger normalizes texts and identifies characters with the *nuqtaa*, it automatically deletes the *nuqtaa* from the characters –

[7] X is a variable of the type of compound. See Bharati et al (2006)

which forces users to search words without a *nuqtaa*: ex. पीड़ा → पीडा. This means that original texts would be missing and never reappear after being tagged. Moreover, if the characters remain as they are, without a *nuqtaa,* it can cause a problem when searching for words with a *nuqtaa*. To avoid this, we have devised the following:

(1) Text normalization 1: Concatenated character string → Combined character
    Ex. ड़ (095C) → ड़ (095C), ड़ (0921+093C) → ड़ (095C)
(2) Text normalization 2: Deleting the *nuqtaa* from concatenated strings too before tagging

As mentioned above, combined characters such as ड़ (095C) in original texts were being converted into characters without *nuqtaa,* such as ड (0921); as in पीडा (surface form, not lemma). Therefore, we added a pre-treatments to the general normalization process. Firstly, we kept the combined characters as they are, and converted the concatenated character strings into the combined characters – this in order to keep the original texts. Secondly, we deleted the *nuqtaa* from concatenated strings too before tagging - this because the tagger tends to delete *nuqtaa* only from combined characters, not from concatenated character strings. Technically, the tagger replaces, e.g., ड़ (095C) with ड (0921). We merged the outputs tagged by the tagger with the changed texts made in processes (1) and (2). An illustrative example follows.

surface form	lemma	tag	details	POS	gender	number	person	case
पीड़ा	पीडा	NN	0	n	f	sg	3	d

We can see *पीड़ा* in the surface form column. It contains the character ड़ (095C). This pre-treatment allowed us to do a uniform search by either the combined characters or concatenated characters at the level of surface forms, and to keep the original texts as they are.

3.4 Development of a specific concordancer for linguistic research

We developed a concordancer to run searches on COSH. This is a web application. A search request made by the user goes through a web framework called Django, and the search is done on a BlackLab server. The search result returns to Django and is displayed on the interface.

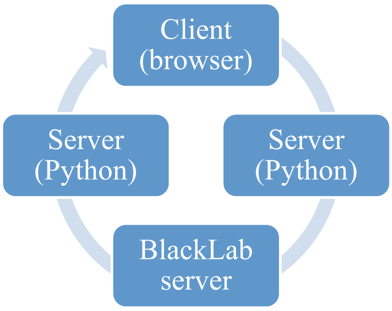


Figure 1

Every time we run a search on the COSH, the client browser requests it from the BlackLab server. Then the Python system we developed processes the search results on the server side, and eventually displays them on the browser, i.e., client side.

4 What kind of linguistic research can we do with the corpora and concordancer?

Although, to my knowledge, there have been few linguistic studies on Hindi-Urdu using a corpus thus far, it's possible to say that this kind of corpora enables even us non-native speakers, who lack intuition of the Hindi language, to do linguistic research on semantic and pragmatic levels. Here's an introduction of the kind of studies we can do using this corpus.

4.1 Verb (V1) + Verb (V2) concatenation

It is well known that Hindi has V1 (a main verb) + V2 (an auxiliary verb) device for adding nuances of lexical aspect or modality to the main verb. Masica (1991) labels meanings expressed by V2s as *Aktionsart,* a term derived from Germanic linguistics.

We can observe a rather similar device in Japanese. Of the V2s used in Japanese, the verb *shimau* 'PUT something AWAY' or 'finish' is frequently used to nuance the V1 in the *-te* form: a conjunctive participle quite similar to *jaa-naa* in Hindi. *Shimau* essentially adds a nuance of 'completeness' to the meaning of the V1, which relates to lexical aspect. The nuance of 'completeness' added by *shimau* is sometimes extended to a modality such as unconsciousness, non-intentionality, and even regret depending on the given context.

Even though they have different meanings respectively, both *jaa-naa* and *shimau* behave alike by adding nuance to a V1's meaning. Both of their nuances depend on the given context, which is a good reason to find out what *kinds* of nuance V2s can add to V1s. Technically, the biggest question from a viewpoint of universal grammar is what exactly the compound verbs are. However, this is a pragmatic issue that native speakers have little consciousness of, and thus is difficult to explain to non-native speakers. Since natives have already learnt how to use the target language unconsciously, we naturally find that different informants often explain different impressions for the nuance, which rather confuses non-native speakers.

What, then, can we non-native speakers do to understand what exactly the compound verbs are? One key method is an investigation of real behaviours of V2 intensively and collectively using a large-scale corpus. Specifically, we can check how frequently those V2s are used, in what context and environment, if there are any restrictions when using them, and in what genres they are most frequently used. These aspects are all noticed by non-native speakers, and not by native speakers who care little for them when using the language.

For example, we investigated restrictions on the co-occurrence in Hindi of the STEM form of the main verb plus the vector or auxiliary verb *jaa-naa* 'GO' together with negative markers, using a Hindi corpus (Nishioka 2015). On this point, Jagannathan (1981: 272-3) claims that the Hindi negative markers such as *nahiiN* do not occur with a 'coloring verb', i.e., a secondary verb (V2) in a verb-verb concatenation. Snell (2010: 290), possibly in support of this claim, explains that "compound verbs give a specific sense of the way in which a particular action is done. It therefore follows that a sentence that's negative or general won't use them; …"

How about Japanese, then? There are numerous studies on *hukugo-doshi*, i.e., compound verb(s). Many of these are limited to explanations to native speakers, except, e.g., Teramura (1984) and Himeno (1999). Recently, Kageyama (2013) began to provide a Compound Verb Lexicon[8]. However, before Nishioka (2013), there seems to have been no specific study using a large corpus[9] that tries to point out why V2s do not occur in negative sentences and to clarify the relation between V2s and negative sentences. This is natural, since this is a matter that non-native speakers easily find when learning the target language.

As we see, using corpora offers the following benefits: non-native speakers of the target language can check a linguistic phenomenon or fact of the target language as objectively and quantitatively as possible; and we can observe the phenomenon from various aspects as required, since COSH provides a context-reference function around the example we have searched.

[8] The site is available at http://vvlexicon.ninjal.ac.jp/en/.

[9] The BCCWJ corpus, provided by National Institute for Japanese Language and Linguistics (NINJAL).

4.2 Noun modification and nominalization

Both Hindi and Japanese are SOV and head-final languages, although from different language families. In Hindi, there are four ways of noun modification and nominalization. The most notable way is modifications of a relative clause or an appositive clause. The other three are genitive postposition *kaa* with the allomorphs *ke* and *kii; vaalaa* with the allomorphs *vale* and *vaalii*, depending on the number and gender of the following noun; and imperfect/perfect participles.

Japanese also has two ways of noun modification and nominalization. One is the genitive case particle *no* (*kaku-joshi* in traditional Japanese language study) also considered a 'quasi-nominal particle' (*juntai-joshi*, considered a functional subset of the *kaku-joshi*); and the other is imperfect/perfect participles. In particular, the particle *no* is said to have multiple functions. For example, Wrona (2012) has summarized the functions of *no* throughout the history of Japanese: Copula (Adnominal), Genitive, Subjective marker, Pronominal, Complementizer, Stance-marker 1, and Stance-marker 2.

In fact, the participial modification in Japanese basically corresponds to modification of a relative clause or participles in Hindi. However, regarding the Hindi connection of Noun 1 and Noun 2 (the latter being a head noun), there are two devices: *kaa* and *vaalaa*. As for the latter, Kellogg (1876: 252, 317) and Beams (1879: 238-9) explain that *vaalaa* was descended from the Sanskrit *paalaka* 'keeper, protector'. Etymologically, it appears to have been used for forming nouns of agency. Although there seem to be no studies on the historical development of the functions of *vaalaa*, it must have developed other functions subsequent to its original etymology. In any event, there is a possibility that these two devices share the respectively different functions of noun modification and even nominalization, as seen in Japanese.

Under the circumstances, what contribution might a large corpus such as COSH make to linguistic studies? With this corpus, we can observe instances of actual use, based on the word itself or combinations of other POS and the word. For example, we find certain noun phrases, such as *piine kaa paanii* [drink.INF.OBL GEN water] and *piine vaalaa paanii* [drink.INF.OBL *vaalaa* water], both of which mean 'water to drink'; or *chuTTii ke din* [holiday GEN din] and *chuTTii vale din* [holiday *vaalaa* day] 'on a holiday'. However, these seem to be used in slightly different contexts. We can also find other examples with GEN or with *vaalaa*. Large corpora like COSH allow us to do a search easily, and to see a context around the example. Moreover, we can search the corpora specifying a part of speech. The corpora allow us to set an infinitive oblique form [*-ne*] in the slot for Noun 1, should we need to limit ourselves to examples with only infinitive forms in that slot.

5 Conclusion

While the use of large corpora is not yet popular in South Asian language research, it is possible to say that the spread such use can encourage us non-native speakers to investigate linguistic phenomena more deeply than before, especially from viewpoints of pragmatics and semantics – that is, to pursue usage-based studies. Although, as we've seen in section 3, we have some points to improve in text annotation in the corpus, COSH will provide powerful supporting evidence to compensate for lack of intuition of the target language in linguistic research by non-native speakers.

Although the scope of this paper did not permit us to include many supporting examples of the aspects of language research, we hope this corpus study will contribute to the pragmatic and semantic study of the Hindi language by non-native speakers.

Acknowledgment

This work was supported by grants-in-aid for scientific research (C) [JSPS KAKENHI: no. JP15K02517 and no. JP20520384] and a grant-in-aid for challenging exploratory research [JSPS KAKENHI: no. JP23652084].

Reference

Beams, John. (1879). *Comparative Grammar of the Modern Aryan Languages of India : to wit, Hindi, Panjabi, Sindhi, Gujarati, Marathi, Oriya and Bangali* (Reprinted by Cambridge University Press edition, Volume 3: The Verb). New York: Trübner, reprinted by Cambridge University Press, 2012.

Bharathi, Akshar, Sharma, Dipi Mishra, Bai, Lakshmi and Sangal, Rajeev (2006). "Ann Corra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages", Language Technologies Research Center, International Institute of Information Technology, IIT Hyderabad, India.

Bharathi, Akshar and Mannem, Prashanth R. (2007). "Introduction to the Shallow Parsing Contest for South Asian Languages", Language Technologies Research Center, International Institute of Information Technology, Hyderabad, India 500032.

Dalal, Aniket, Nagaraj, Kumar, Sawant, Uma, Shelke, Sandeep, and Bhattacharyya, Pushpak. (2007). "Building Feature Rich POS Tagger for Morphologically Rich Languages: Experiences in Hindi". In *Proceedings of ICON*. Available from https://www.cse.iitb.ac.in/~pb/papers/icon07-Hindi-memm-postag.pdf

Himeno, Masako (1999). *Hukugou doushi no imi youhou to kouzou*. Tokyo: Hitsuji shobou.

Hook, Peter E. (1974). *The Compound Verb in Hindi.* Michigan: University of Michigan, Center for South and Southeast Asian Studies.

Ishikawa, Shinichiro, Maeda, Tadahiko, and Yamazaki, Makoto (eds.) (2010) *Gengo Kenkyuu no tame no Toukei Nyuumon*.Tokyo: Kuroshio shuppan.

Ishikawa, Shinichiro (2012) *Basic Corpus Gengogaku (A Basic Guide to Corpus Linguistics)*. Tokyo: Hitsuji shobou.

Jagannaathan, V. R. (1981). *Prayog aur prayog.* Dillii: Oxford University Pres.

Kachru, Yamuna. (1980). *Aspects of Hindi Grammar*. New Delhi: Manohar.

—— (2006). *Hindi*. Amsterdam/Philadelphia: John Benjamins Pub Co.

Kellogg, H. Samuel. (1938). *A Grammar of the Hindi Language : in which are treated the High Hindí, Braj, and the Eastern Hindí of the Rámáyan of Tulsí Dás, also the colloquial dialects of Rájputáná, Kumáon, Avadh, Ríwá, Bhojpúr, Magadha, Maithila, etc., with copious philological note (the 3rd edition)*. London: Kegan Paul, Trench, Trubner and Co.

Masica, Colin. P. (1976). *Defining a Linguistic Area: South Asia.* Chicago: University of Chicago Press.

—— (1991). *The Indo-Aryan Languages*. Cambridge: Cambridge University Press.

McGregor, R.S. (1995). *Outline of Hindi Grammar*. Oxford: Oxford University Press.

Nishioka, Miki. (2005). "Hindii-go no iwayu ru meishiku ni tsuite: zokkaku kouchishi '*kaa*' wo chuushin-ni", *Kyoto Sangyo University essays Humanities series vol.33*. Pp. 74-98. Kyoto: Kyoto Industrial University.

—— (2013). "Te-kei + *shimau* to hiteiji to no kyouki seigen to sono kankyou ni tsuite: Hindii-go to no taishougengogakuteki shiten kara (Co-occurrence Restrictions on the '*-te* Form + *shimau*' and Negation in Japanese: A Contrastive Analysis with Hindi)", *Matani ronshū vol.7*. Pp.47-73, Osaka: Nihongo Nihon Bunka Kyouiku Kenkyuukai.

—— (2014). "Co-occurrence restrictions on the '*-te* Form + *shimau*' and negation in Japanese: A contrastive analysis with Hindi", presented at XXVIIes Journées de Linguistique d' Asie Orientale. Abstract is available from http://crlao.ehess.fr/docannexe/file/1698/booklet.pdf.

—— (2016) "Functions of *jaanaa* as a V2 in Hindi: From Lexicalization to Grammaticalization", presented at 32nd South Asian Languages Analysis Roundtable (SALA-32). Abstract (pp.57-9) is available from http://media.wix.com/ugd/56a455_223031a3531f4b18b7ad857a6626cc7b.pdf

Nishioka, Miki and Akasegawa, Shiro (2015) "Restrictions on co-occurrence of 'STEM + *jaanaa*' and negation in Hindi: a contrastive analysis with '*-te* + *shimau*' in Japanese". In Book of Abstracts South Asian Languages Analysis Roundtable (SALA 31). Abstract (pp.51-3) is available from http://ucrel.lancs.ac.uk/sala-31/doc/ABSTRACTBOOK-maincontent.pdf

Noonan, Michael. (1997). "Versatile Nominalizations". Bybee, Joan, Haiman, John, Thompson, Sandra A. (eds.), *Essays on Language Function and Language Type: Dedicated to T. Givón*. Pp. 373-394. John Benjamins Publishing.

P.J, Antony and K.P., Sonam. (2011). “Part of Speech Tagging for Indian Languages: A Literature Survey”. In *International Journal of Computer Applications (0975-8887)*, Volume 34, No.8, pp.22-9. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.1220&rep=rep1&type=pdf

Snell, Rupert (2010). *Teach Yourself Get Started in Hindi (Teach Yourself Beginner's Languages)*. 2nd Revised. Teach Yourself Books.

Teramura, Hideo. (1984). *Nihongo no syntax to imi*, vol II (11th edition). Tokyo: Kuroshio shuppan.

Wrona, Janick. (2012). “The Early History of no as a Nominaliser”. Frellesvig, Bjarke, Kiaer, Jieun, Wrona, Janick (eds.), *Studies in Asian Linguistics (LSASL 78): Studies in Japanese and Korean Linguistics*, available from http://www.engl.polyu.edu.hk/research/nomz/pdf/WRONA_History_of_NO.pdf#search='The+Early+History+of+no+as+a+Nominaliser'). München: LINCOM.