Hindi Word Sketches

Anil Krishna Eragani[1], Varun Kuchibhotla[1], Dipti Misra Sharma[1], Siva Reddy[2], and Adam Kilgarriff[3]
[1]IIIT Hyderabad, India; [2]University of Edinburgh; [3]UK, Lexical Computing Ltd., UK
{anil.eragani,varun.k}@research.iiit.ac.in, dipti@iiit.ac.in,
siva.reddy@ed.ac.uk, adam@lexmasterclass.com

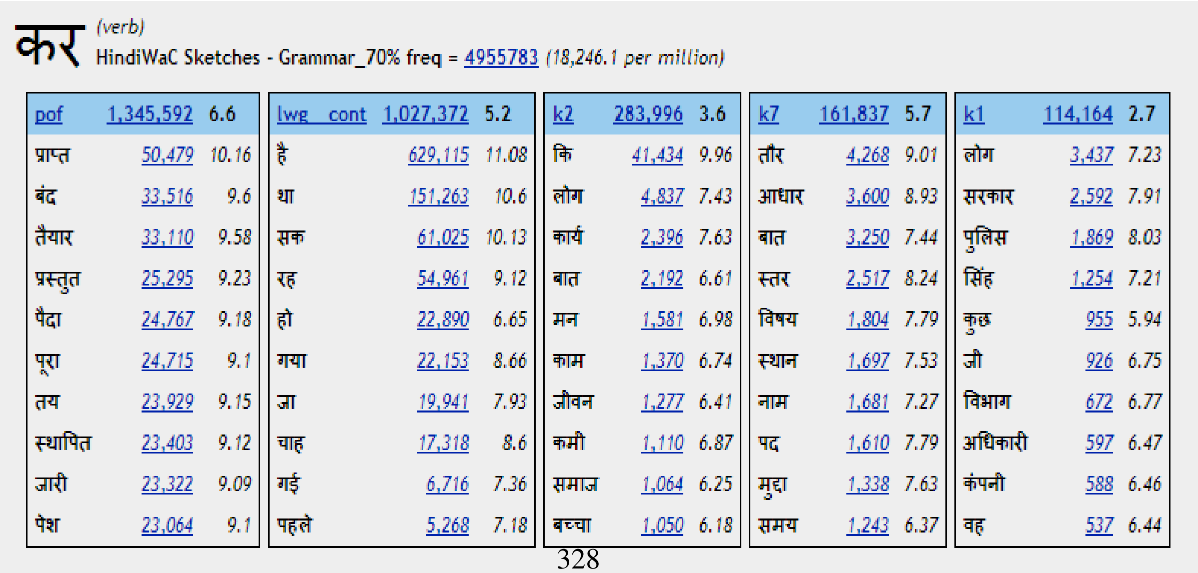
Abstract

Word sketches are one-page automatic, corpus-based summaries of a word's grammatical and collocational behaviour. These are widely used for studying a language and in lexicography. Sketch Engine is a leading corpus tool which takes as input a corpus and generates word sketches for the words of that language. It also generates a thesaurus and 'sketch differences', which specify similarities and differences between near-synonyms. In this paper, we present the functionalities of Sketch Engine for Hindi. We collected HindiWaC, a web crawled corpus for Hindi with 240 million words. We lemmatized, POS tagged the corpus and then loaded it into Sketch Engine.

1 Introduction

A language *corpus* is simply a collection of texts, so-called when it is used for language research. Corpora can be used for all sorts of purposes: from literature to language learning; from discourse analysis to grammar to language change to sociolinguistic or regional variation; from translation to technology.

Corpora are becoming more and more important, because of computers. On a computer, a corpus can be searched and explored in all sorts of ways. Of course that requires the right app. One leading app for corpus querying is the Sketch Engine (Kilgarriff et al., 2004). The Sketch Engine has been in daily use for writing dictionary entries since 2004, first at Oxford University Press, more recently at Cambridge University Press, Collins, Macmillan, and in National Language Institutes for Czech, Dutch, Estonian, Irish, Slovak and Slovene. It is also in use for all the other purposes listed above. On logging in to the Sketch Engine, the user can explore corpora for sixty languages. In many cases the corpora are the largest and best available for the language. For Indian languages, there are the corpora for Bengali, Gujarati, Hindi, Malayalam, Tamil and Telugu. The largest is for Hindi with

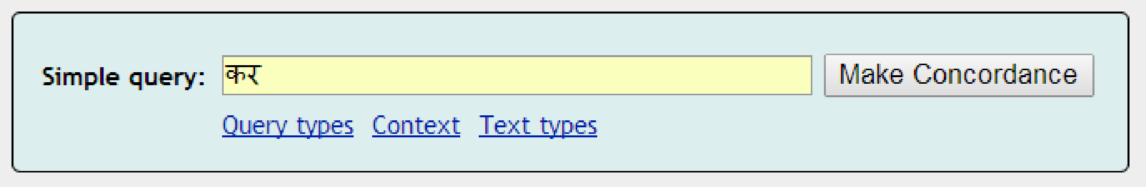


कर (verb)
HindiWaC Sketches - Grammar_70% freq = 4955783 (18,246.1 per million)

pof	1,345,592	6.6	lwg_cont	1,027,372	5.2	k2	283,996	3.6	k7	161,837	5.7	k1	114,164	2.7
प्राप्त	50,479	10.16	है	629,115	11.08	कि	41,434	9.96	तौर	4,268	9.01	लोग	3,437	7.23
बंद	33,516	9.6	था	151,263	10.6	लोग	4,837	7.43	आधार	3,600	8.93	सरकार	2,592	7.91
तैयार	33,110	9.58	सक	61,025	10.13	कार्य	2,396	7.63	बात	3,250	7.44	पुलिस	1,869	8.03
प्रस्तुत	25,295	9.23	रह	54,961	9.12	बात	2,192	6.61	स्तर	2,517	8.24	सिंह	1,254	7.21
पैदा	24,767	9.18	हो	22,890	6.65	मन	1,581	6.98	विषय	1,804	7.79	कुछ	955	5.94
पूरा	24,715	9.1	गया	22,153	8.66	काम	1,370	6.74	स्थान	1,697	7.53	जी	926	6.75
तय	23,929	9.15	जा	19,941	7.93	जीवन	1,277	6.41	नाम	1,681	7.27	विभाग	672	6.77
स्थापित	23,403	9.12	चाह	17,318	8.6	कमी	1,110	6.87	पद	1,610	7.79	अधिकारी	597	6.47
जारी	23,322	9.09	गई	6,716	7.36	समाज	1,064	6.25	मुद्दा	1,338	7.63	कंपनी	588	6.46
पेश	23,064	9.1	पहले	5,268	7.18	बच्चा	1,050	6.18	समय	1,243	6.37	वह	537	6.44

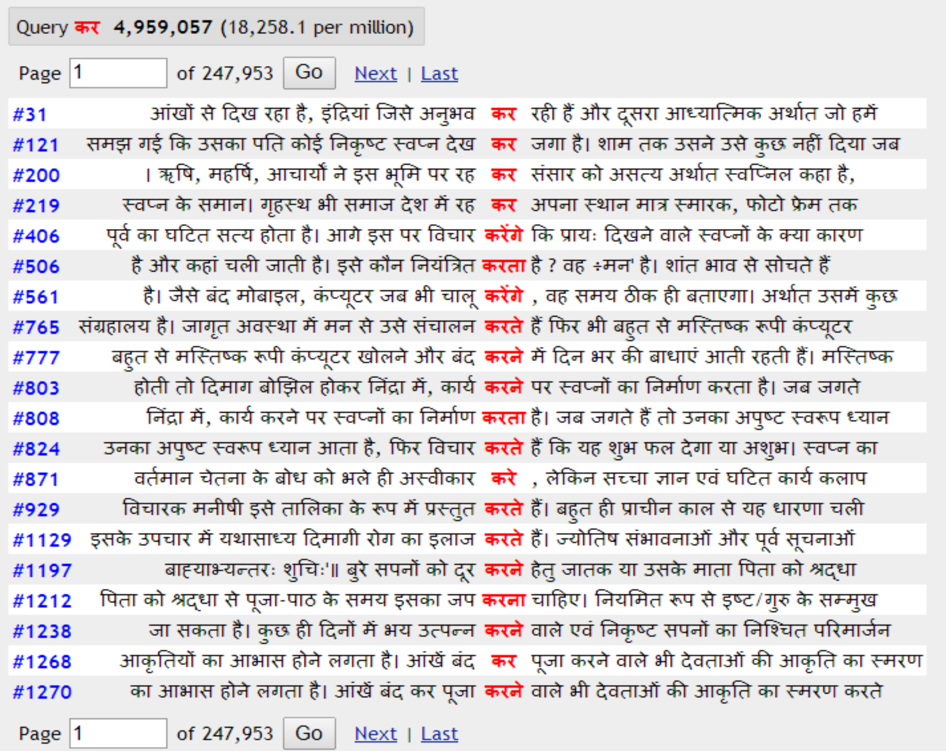
I

Figure 1. Word sketches for the verb **कर** (do)



Simple query: कर

Figure 2. Simple concordance query



Query **कर** **4,959,057** (18,258.1 per million)

Page 1 of 247,953

#31	आंखों से दिख रहा है, इंद्रियां जिसे अनुभव	**कर**	रही हैं और दूसरा आध्यात्मिक अर्थात जो हमें
#121	समझ गई कि उसका पति कोई निकृष्ट स्वप्न देख	**कर**	जगा है। शाम तक उसने उसे कुछ नहीं दिया जब
#200	। ऋषि, महर्षि, आचार्यों ने इस भूमि पर रह	**कर**	संसार को असत्य अर्थात स्वप्निल कहा है,
#219	स्वप्न के समान। गृहस्थ भी समाज देश में रह	**कर**	अपना स्थान मात्र स्मारक, फोटो फ्रेम तक
#406	पूर्व का घटित सत्य होता है। आगे इस पर विचार	**करेंगे**	कि प्रायः दिखने वाले स्वप्नों के क्या कारण
#506	है और कहां चली जाती है। इसे कौन नियंत्रित	**करता**	है ? वह ÷मन' है। शांत भाव से सोचते हैं
#561	है। जैसे बंद मोबाइल, कंप्यूटर जब भी चालू	**करेंगे**	, वह समय ठीक ही बताएगा। अर्थात उसमें कुछ
#765	संग्रहालय है। जागृत अवस्था में मन से उसे संचालन	**करते**	हैं फिर भी बहुत से मस्तिष्क रूपी कंप्यूटर
#777	बहुत से मस्तिष्क रूपी कंप्यूटर खोलने और बंद	**करने**	में दिन भर की बाधाएं आती रहती हैं। मस्तिष्क
#803	होती तो दिमाग बोझिल होकर निंद्रा में, कार्य	**करने**	पर स्वप्नों का निर्माण करता है। जब जगते
#808	निंद्रा में, कार्य करने पर स्वप्नों का निर्माण	**करता**	है। जब जगते हैं तो उनका अपुष्ट स्वरूप ध्यान
#824	उनका अपुष्ट स्वरूप ध्यान आता है, फिर विचार	**करते**	हैं कि यह शुभ फल देगा या अशुभ। स्वप्न का
#871	वर्तमान चेतना के बोध को भले ही अस्वीकार	**करे**	, लेकिन सच्चा ज्ञान एवं घटित कार्य कलाप
#929	विचारक मनीषी इसे तालिका के रूप में प्रस्तुत	**करते**	हैं। बहुत ही प्राचीन काल से यह धारणा चली
#1129	इसके उपचार में यथासाध्य दिमागी रोग का इलाज	**करते**	हैं। ज्योतिष संभावनाओं और पूर्व सूचनाओं
#1197	बाह्याभ्यन्तरः शुचिः'॥ बुरे सपनों को दूर	**करने**	हेतु जातक या उसके माता पिता को श्रद्धा
#1212	पिता को श्रद्धा से पूजा-पाठ के समय इसका जप	**करना**	चाहिए। नियमित रूप से इष्ट/गुरु के सम्मुख
#1238	जा सकता है। कुछ ही दिनों में भय उत्पन्न	**करने**	वाले एवं निकृष्ट सपनों का निश्चित परिमार्जन
#1268	आकृतियों का आभास होने लगता है। आंखें बंद	**कर**	पूजा करने वाले भी देवताओं की आकृति का स्मरण
#1270	का आभास होने लगता है। आंखें बंद कर पूजा	**करने**	वाले भी देवताओं की आकृति का स्मरण करते

Page 1 of 247,953

Figure 3. The resulting concordance lines

240 million words – we would be referring to it as HindiWaC in the rest of the paper.

The function that gives the Sketch Engine its name is the **'word sketch'**, a one-page, automatically-derived summary of a word's grammatical and collocational behaviour, as in Figure 1. Since the images in this paper are screen shots taken from Sketch Engine, translations and gloss have not been provided for the Hindi words in the images.

In this paper we first introduce the main functions of the Sketch Engine, with Hindi examples. We then describe how we built and processed HindiWaC, and set it up in the Sketch Engine.

2 The Sketch Engine For Hindi

2.1 The Simple Concordance Query Function

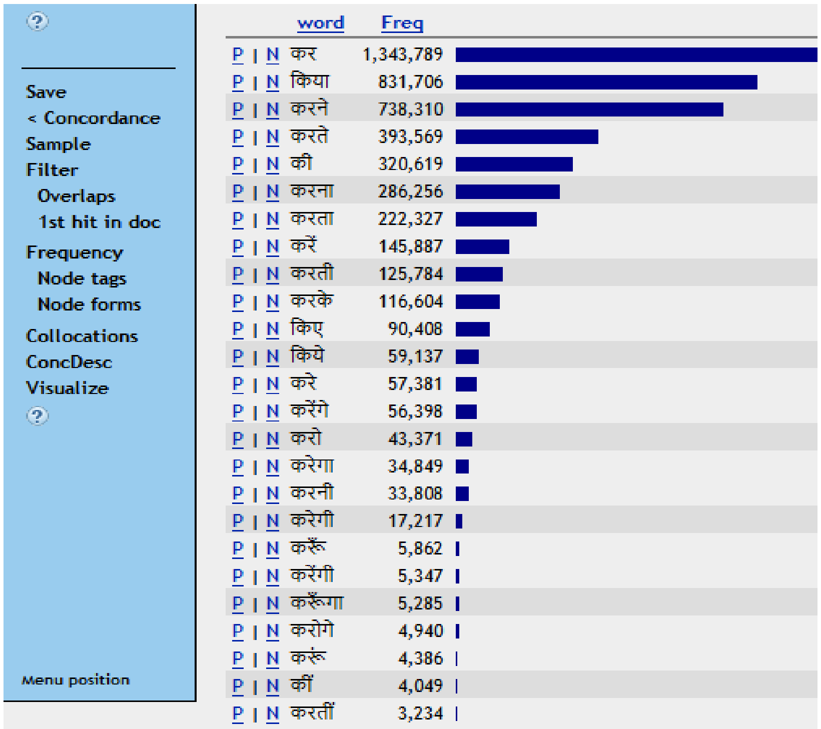
A Simple concordance query shows the word as it is used in different texts. Figure 2 shows the query box, while Figures 3 shows its output. A simple search query for a word such as कर (do) searches for the lemma as well as the words which have कर (do) as the lemma, so कर (do), किया (did), करने (to do), करते ([they] will do), etc. are all retrieved. Figure 3 shows the first 20 results out of the retrieved ~5 million results.

2.2 The Frequency Functions

The Sketch Engine interface provides easy access to tools for visualizing different aspects of the word frequency (see Figure 4). The *Frequency Node forms* function on the left hand menu in Figure 4 shows which of the returned forms are most frequent.

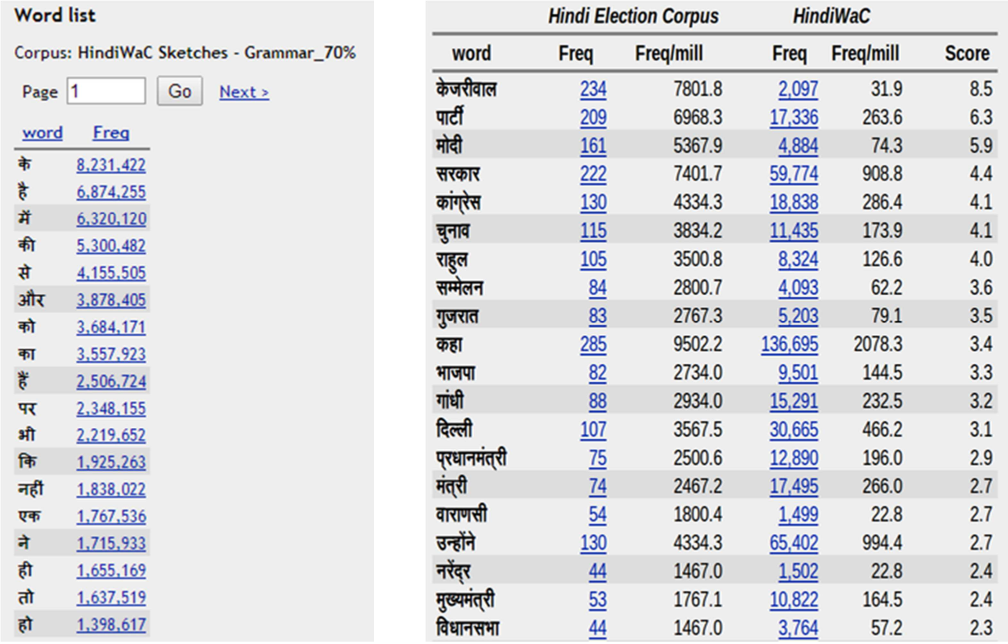
Thus we have immediately discovered that the commonest forms of the lemma कर (do) are कर (do), किया (did) and करने (to do).

The **p/n** links are for positive and negative examples. Clicking on **p** gives a concordance for the word form, while clicking on **n** gives the whole concordance *except* for the word form.



	word	Freq
P | N	कर	1,343,789
P | N	किया	831,706
P | N	करने	738,310
P | N	करते	393,569
P | N	की	320,619
P | N	करना	286,256
P | N	करता	222,327
P | N	करें	145,887
P | N	करती	125,784
P | N	करके	116,604
P | N	किए	90,408
P | N	किये	59,137
P | N	करे	57,381
P | N	करेंगे	56,398
P | N	करो	43,371
P | N	करेगा	34,849
P | N	करनी	33,808
P | N	करेगी	17,217
P | N	करूँ	5,862
P | N	करेंगी	5,347
P | N	करूँगा	5,285
P | N	करोगे	4,940
P | N	करूं	4,386
P | N	कीं	4,049
P | N	करतीं	3,234

Figure 4. Frequency of word forms of कर (do)



Word list

Corpus: HindiWaC Sketches - Grammar_70%

word	Freq
के	8,231,422
है	6,874,255
में	6,320,120
की	5,300,482
से	4,155,505
और	3,878,405
को	3,684,171
का	3,557,923
हैं	2,506,724
पर	2,348,155
भी	2,219,652
कि	1,925,263
नहीं	1,838,022
एक	1,767,536
ने	1,715,933
ही	1,655,169
तो	1,637,519
हो	1,398,617

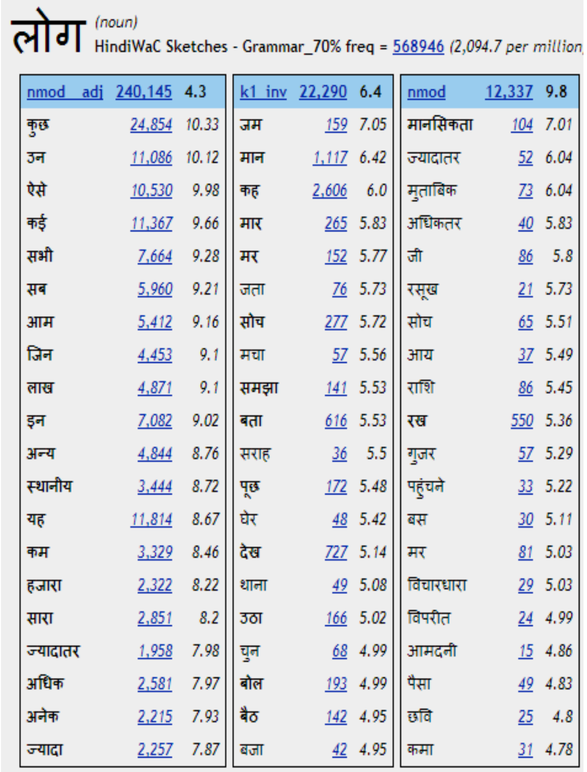
	Hindi Election Corpus		*HindiWaC*		
word	Freq	Freq/mill	Freq	Freq/mill	Score
केजरीवाल	234	7801.8	2,097	31.9	8.5
पार्टी	209	6968.3	17,336	263.6	6.3
मोदी	161	5367.9	4,884	74.3	5.9
सरकार	222	7401.7	59,774	908.8	4.4
कांग्रेस	130	4334.3	18,838	286.4	4.1
चुनाव	115	3834.2	11,435	173.9	4.1
राहुल	105	3500.8	8,324	126.6	4.0
सम्मेलन	84	2800.7	4,093	62.2	3.6
गुजरात	83	2767.3	5,203	79.1	3.5
कहा	285	9502.2	136,695	2078.3	3.4
भाजपा	82	2734.0	9,501	144.5	3.3
गांधी	88	2934.0	15,291	232.5	3.2
दिल्ली	107	3567.5	30,665	466.2	3.1
प्रधानमंत्री	75	2500.6	12,890	196.0	2.9
मंत्री	74	2467.2	17,495	266.0	2.7
वाराणसी	54	1800.4	1,499	22.8	2.7
उन्होंने	130	4334.3	65,402	994.4	2.7
नरेंद्र	44	1467.0	1,502	22.8	2.4
मुख्यमंत्री	53	1767.1	10,822	164.5	2.4
विधानसभा	44	1467.0	3,764	57.2	2.3

Figures 5 & 6. Frequency list of the whole corpus for Words and Keywords extracted automatically from Hindi Election Corpus by comparing it with Hindi Web Corpus

2.3 The Word List function

The Word List function allows the user to make frequency lists of many types (words, lemmas, tags). Figure 5 shows the most frequent words in the corpus. In addition to most frequent words, keywords of any target corpus can be extracted. This is done by comparing frequent words from the target corpus with the frequent words from a general purpose corpus. Figure 6 displays the keywords of a Hindi Election corpus, where this is the target corpus, and the general purpose corpus is the HindiWaC.

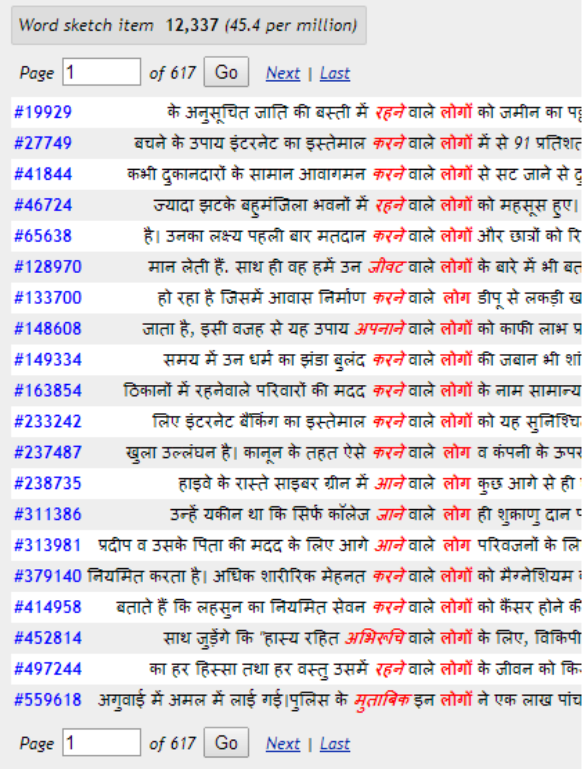
Almost every keyword closely relates to the trend of news articles in the 2014 Indian Parliament elections. Since the Hindi Election Corpus is of small size, the frequency-per-million column contains projected values. These are signifi-



लोग (noun)
HindiWaC Sketches - Grammar_70% freq = 568946 (2,094.7 per million

nmod_adj	240,145	4.3	k1_inv	22,290	6.4	nmod	12,337	9.8
कुछ	24,854	10.33	जम	159	7.05	मानसिकता	104	7.01
उन	11,086	10.12	मान	1,117	6.42	ज्यादातर	52	6.04
ऐसे	10,530	9.98	कह	2,606	6.0	मुताबिक	73	6.04
कई	11,367	9.66	मार	265	5.83	अधिकतर	40	5.83
सभी	7,664	9.28	मर	152	5.77	जी	86	5.8
सब	5,960	9.21	जता	76	5.73	रसूख	21	5.73
आम	5,412	9.16	सोच	277	5.72	सोच	65	5.51
जिन	4,453	9.1	मचा	57	5.56	आय	37	5.49
लाख	4,871	9.1	समझा	141	5.53	राशि	86	5.45
इन	7,082	9.02	बता	616	5.53	रख	550	5.36
अन्य	4,844	8.76	सराह	36	5.5	गुजर	57	5.29
स्थानीय	3,444	8.72	पूछ	172	5.48	पहुंचने	33	5.22
यह	11,814	8.67	घेर	48	5.42	बस	30	5.11
कम	3,329	8.46	देख	727	5.14	मर	81	5.03
हजारा	2,322	8.22	थाना	49	5.08	विचारधारा	29	5.03
सारा	2,851	8.2	उठा	166	5.02	विपरीत	24	4.99
ज्यादातर	1,958	7.98	चुन	68	4.99	आमदनी	15	4.86
अधिक	2,581	7.97	बोल	193	4.99	पैसा	49	4.83
अनेक	2,215	7.93	बैठ	142	4.95	छवि	25	4.8
ज्यादा	2,257	7.87	बजा	42	4.95	कमा	31	4.78

Figure 7. Word Sketch results for लोग (people)



Word sketch item 12,337 (45.4 per million)

Page 1 of 617 Go Next | Last

#19929	के अनुसूचित जाति की बस्ती में *रहने* वाले	लोगों	को जमीन का प
#27749	बचने के उपाय इंटरनेट का इस्तेमाल *करने* वाले	लोगों	में से 91 प्रतिश
#41844	कभी दुकानदारों के सामान आवागमन *करने* वाले	लोगों	से सट जाने से दु
#46724	ज्यादा झटके बहुमंजिला भवनों में *रहने* वाले	लोगों	को महसूस हुए।
#65638	है। उनका लक्ष्य पहली बार मतदान *करने* वाले	लोगों	और छात्रों को रि
#128970	मान लेती हैं. साथ ही वह हमें उन *जीवट* वाले	लोगों	के बारे में भी बत
#133700	हो रहा है जिसमें आवास निर्माण *करने* वाले	लोग	डीपू से लकड़ी ख
#148608	जाता है, इसी वजह से यह उपाय *अपनाने* वाले	लोगों	को काफी लाभ प्र
#149334	समय में उन धर्म का झंडा बुलंद *करने* वाले	लोगों	की जबान भी शां
#163854	ठिकानों में रहनेवाले परिवारों की मदद *करने* वाले	लोगों	के नाम सामान्य
#233242	लिए इंटरनेट बैंकिंग का इस्तेमाल *करने* वाले	लोगों	को यह सुनिश्चि
#237487	खुला उल्लंघन है। कानून के तहत ऐसे *करने* वाले	लोग	व कंपनी के ऊपर
#238735	हाइवे के रास्ते साइबर ग्रीन में *आने* वाले	लोग	कुछ आगे से ही
#311386	उन्हें यकीन था कि सिर्फ कॉलेज *जाने* वाले	लोग	ही शुक्राणु दान प
#313981	प्रदीप व उसके पिता की मदद के लिए आगे *आने* वाले	लोग	परिवजनों के लि
#379140	नियमित करता है। अधिक शारीरिक मेहनत *करने* वाले	लोगों	को मैग्नेशियम
#414958	बताते हैं कि लहसुन का नियमित सेवन *करने* वाले	लोगों	को कैंसर होने कि
#452814	साथ जुड़ेंगे कि "हास्य रहित *अभिरुचि* वाले	लोगों	के लिए, विकिपी
#497244	का हर हिस्सा तथा हर वस्तु उसमें *रहने* वाले	लोगों	के जीवन को कि
#559618	अगुवाई में अमल में लाई गई।पुलिस के *मुताबिक* इन	लोगों	ने एक लाख पांच

Page 1 of 617 Go Next | Last

Figure 8. Concordance lines for लोग (people) in combination with its gramrel "nmod"

cantly higher than the same words in the HindiWaC since the Election Corpus is domain specific.

2.4 The Word Sketch and Collocation Concordance functions

The Word Sketch function is invaluable for finding collocations. The word sketches of the word लोग (people) for three dependency relations are shown in Figure 7.

The dependency relations that we use are based on the Paninian framework (Begum et al., 2008). Three of the most common dependency relations given by this model are as follows:

- k1: agent and/or doer
- k2: object and/or theme
- k3: instrument

These relations are syntactico-semantic in nature, and differ slightly from the equivalent thematic roles mentioned above. More about how we get the word sketches shown in Figure 7 is explained in Section 3.

In figure 7, the three dependency relations shown are:

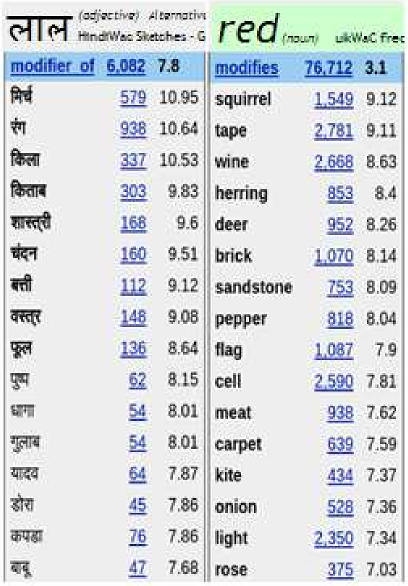
- nmod_adj: noun-modifier_adjective
- k1_inv: doer_inverse
- nmod: noun-modifier

The word sketch function assigns weights to each of the collocates and also to the dependency relations.

Clicking on the number after the collocate gives a concordance of the combination (Figure 8).

2.5 The Bilingual Word Sketch function

A new function has been added recently to the Word Sketch, which is the Bilingual Word Sketch. This allows the user to see word sketches for two words side by side in different languages. Figure 9 shows a comparison between लाल (red) and *red*. Interestingly, the usage of word red in Hindi and English are very diverse. The only common noun which is modified by *red* in both languages is गुलाब (*rose*).



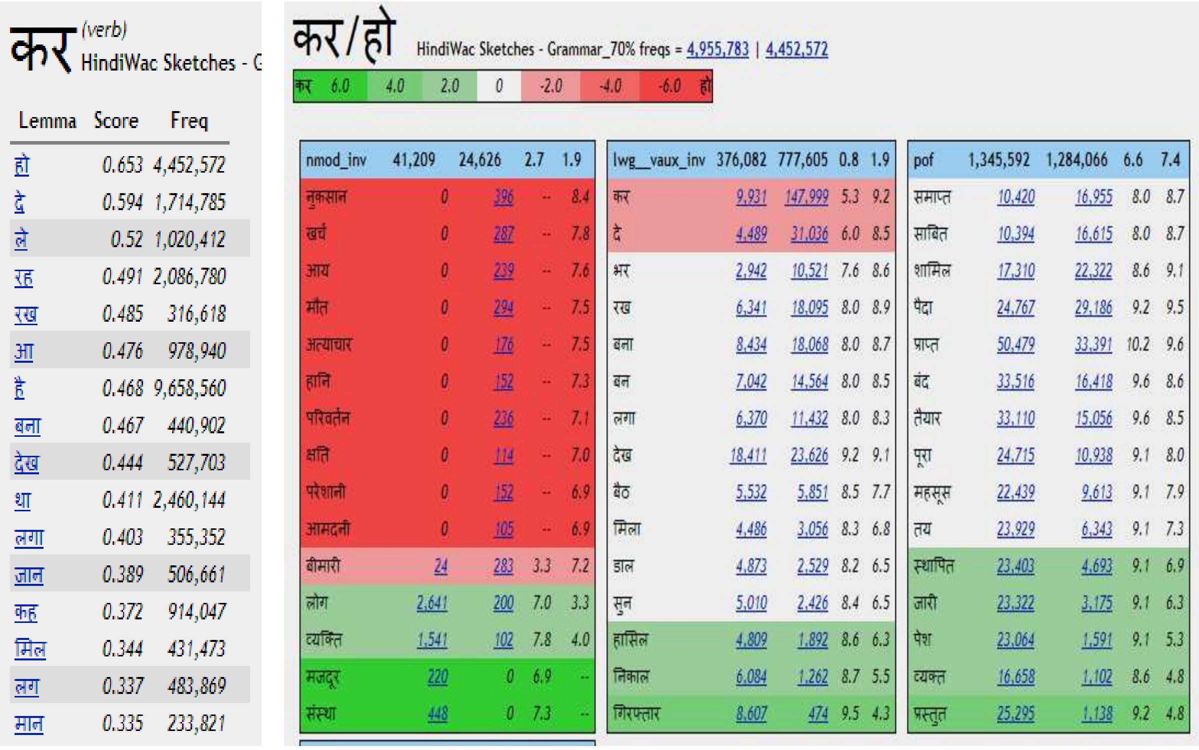
लाल (adjective) Alternativ HindiWac Sketches - G | red (noun) ukWaC Freq

modifier_of	6,082	7.8	modifies	76,712	3.1
मिर्च	579	10.95	squirrel	1,549	9.12
रंग	938	10.64	tape	2,781	9.11
किला	337	10.53	wine	2,668	8.63
किताब	303	9.83	herring	853	8.4
शास्त्री	168	9.6	deer	952	8.26
चंदन	160	9.51	brick	1,070	8.14
बत्ती	112	9.12	sandstone	753	8.09
वस्त्र	148	9.08	pepper	818	8.04
फूल	136	8.64	flag	1,087	7.9
पुष्प	62	8.15	cell	2,590	7.81
धागा	54	8.01	meat	938	7.62
गुलाब	54	8.01	carpet	639	7.59
यादव	64	7.87	kite	434	7.37
डोरा	45	7.86	onion	528	7.36
कपड़ा	76	7.86	light	2,350	7.34
बाबू	47	7.68	rose	375	7.03

Figure 9. Adjective results of a bilingual word sketch for Hindi लाल (red) and English red
English translations of some of the Hindi words are: chilli, colour, fort, flower, rose, cloth, Shastri

2.6 Distributional Thesaurus and Sketch Diff

The Sketch Engine also offers a distributional thesaurus, where, for the input word, the words 'sharing' most collocates are presented. Figure 10 shows the top entries in similarity to कर (do). The top result is हो (be). Clicking on it takes us to a 'sketch diff', a report that shows the similarities and differences between the two words in Figure 11.



कर (verb) HindiWac Sketches - G

Lemma	Score	Freq
हो	0.653	4,452,572
दे	0.594	1,714,785
ले	0.52	1,020,412
रह	0.491	2,086,780
रख	0.485	316,618
आ	0.476	978,940
है	0.468	9,658,560
बना	0.467	440,902
देख	0.444	527,703
था	0.411	2,460,144
लगा	0.403	355,352
जान	0.389	506,661
कह	0.372	914,047
मिल	0.344	431,473
लग	0.337	483,869
मान	0.335	233,821

कर/हो HindiWac Sketches - Grammar_70% freqs = 4,955,783 | 4,452,572

कर 6.0	4.0	2.0	0	-2.0	-4.0	-6.0 हो

nmod_inv	41,209	24,626	2.7	1.9
नुकसान	0	396	--	8.4
खर्च	0	287	--	7.8
आय	0	239	--	7.6
मौत	0	294	--	7.5
अत्याचार	0	176	--	7.5
हानि	0	152	--	7.3
परिवर्तन	0	236	--	7.1
क्षति	0	114	--	7.0
परेशानी	0	152	--	6.9
आमदनी	0	105	--	6.9
बीमारी	24	283	3.3	7.2
लोग	2,641	200	7.0	3.3
व्यक्ति	1,541	102	7.8	4.0
मजदूर	220	0	6.9	--
संस्था	448	0	7.3	--

lwg_vaux_inv	376,082	777,605	0.8	1.9
कर	9,931	147,999	5.3	9.2
दे	4,489	31,036	6.0	8.5
भर	2,942	10,521	7.6	8.6
रख	6,341	18,095	8.0	8.9
बना	8,434	18,068	8.0	8.7
बन	7,042	14,564	8.0	8.5
लगा	6,370	11,432	8.0	8.3
देख	18,411	23,626	9.2	9.1
बैठ	5,532	5,851	8.5	7.7
मिला	4,486	3,056	8.3	6.8
डाल	4,873	2,529	8.2	6.5
सुन	5,010	2,426	8.4	6.5
हासिल	4,809	1,892	8.6	6.3
निकाल	6,084	1,262	8.7	5.5
गिरफ्तार	8,607	474	9.5	4.3

pof	1,345,592	1,284,066	6.6	7.4
समाप्त	10,420	16,955	8.0	8.7
साबित	10,394	16,615	8.0	8.7
शामिल	17,310	22,322	8.6	9.1
पैदा	24,767	29,186	9.2	9.5
प्राप्त	50,479	33,391	10.2	9.6
बंद	33,516	16,418	9.6	8.6
तैयार	33,110	15,056	9.6	8.5
पूरा	24,715	10,938	9.1	8.0
महसूस	22,439	9,613	9.1	7.9
तय	23,929	6,343	9.1	7.3
स्थापित	23,403	4,693	9.1	6.9
जारी	23,322	3,175	9.1	6.3
पेश	23,064	1,591	9.1	5.3
व्यक्त	16,658	1,102	8.6	4.8
प्रस्तुत	25,295	1,138	9.2	4.8

Figure 10 & 11. Thesaurus search showing entries similar to कर (do) (left) and Sketch Diff comparing collocates of कर (do) and हो (be) (right)

The red results occur most frequently with हो (be), the green ones with कर (do). The ones on white occur equally with both.

3 Building and processing HindiWaC and loading it into the Sketch Engine

HindiWaC was built using the Corpus Factory procedure (Kilgarriff et al., 2010). A first tranche was built in 2009, with the crawling process repeated and more data added in 2011, and again in 2014. Corpus Factory method can be briefly described as follows: several thousands of target language search queries are generated from Wikipedia, and are submitted to Microsoft Bing search engine. The corresponding hit pages for each query are downloaded. The pages are filtered using a language model. Boilerplate text is removed using body text extraction, deduplication to create clean corpus. We use jusText and Onion tools (Pomikálek, 2011) for body-text extraction and deduplication tools respectively.

The text is then tokenized, lemmatized and POS-tagged using the tools downloaded from http://sivareddy.in/downloads (Reddy and Sharoff, 2011). The tokenizer found here is installed in the Sketch Engine.

3.1 Sketch Grammar for Hindi

A sketch grammar is a grammar for the language, based on regular expressions over part-of-speech tags. It underlies the word sketches and is written in the Corpus Query Language (CQL). Sketch grammar is designed particularly to identify head-and-dependent pairs of words (e.g., खा [eat] and राम [Ram]) in specified grammatical relations (here, k1 [doer]), in order that the dependent can be entered into the head's word sketch and vice versa.

Sketch Grammars are popular with lexicographic and corpus linguistics community, and are used to identify collocations of a word with a given grammatical relation (Kilgarriff and Rundell, 2002). We use Sketch Grammar to identify words in syntactic relations in a given sentence. For example, a grammar rule for the relation "**k1**" (**doer**) is *2:"NN" "PSP\:ने" "JJ"? 1:"VM"*, which specifies that if a noun is followed by a PSP and an optional adjective and followed by a verb, then the noun is the kartha/subject of the verb. The head and child are identified by 1: and 2: respectively. Each rule may often be matched by more than one relation creating ambiguity. Yet they tend to capture the most common behavior in the language.

Writing a full-fledged sketch grammar with high coverage is a difficult task even for language experts, as it would involve capturing all the idiosyncrasies of a language. Even though such hand-written rules tend to be more accurate, the recall of the rules is very low. In this paper, the grammar we use is a collection of POS tag sequences (rules) which are automatically extracted from an annotated Treebank, Hindi Dependency Treebank (HDT-v0.5), which was released for the Coling2012 shared task on dependency parsing (Sharma et al., 2012). This treebank uses IIIT tagset described in (Bharati et al., 2006). This method gives us a lot of rules based on the syntactic ordering of the words. Though these rules do not have all the lexical cues of a language, the hope is that, when applied on a large-scale web corpus, the correct matches (sketches) of the rules automatically become statistically more frequent, and hence more significant.

From the above mentioned treebank (HDT) we extract dependency grammar rules (i.e. sketch grammar) automatically for each dependency relation, based on the POS tags appearing in between the dependent words (inclusive). For example, from the sentence, राम(Ram) ने(erg.) कमरे(room) में(inside) आम(mango) खाया(eat), Ram ate [a] mango in [the] room, we extract rules of the type: **(k1[doer], k2 [object], k7[location])**

k1 - 2:[tag="NNP"] [tag="PSP\:ने"] [tag="NN"] [tag="PSP\:में"] 1:[tag="VM"]
k2 - 2:[tag="NN"] 1:[tag="VM"]
k7 - 2:[tag="NN"] [tag="PSP\:में"] [tag="NN"] 1:[tag="VM"]

In the above example, relation names are in bold with one of the corresponding rules for each of them.

We do include a few lexical features associated with the POS tags **PSP** (post-position) and **CC** (conjunction) in order to disambiguate between different dependency relations. For example, in both the relations **k1 (doer)** and **k2 (object)** we have the rules (1) and (2) given below respectively in Figure 12:

बच्चे (NN)	ने (PSP)	आम (NN)	खाया (VM)	
(child)	(erg.)	(mango)	(eat.pst)	(the child ate a mango)
2:[tag="NN"]	[tag="PSP\:ने"]	[tag=”NN”]	1:[tag="VM"]	------- (1)
कागज़ (NN)	को (PSP)	बाहर (NN)	फेंका (VM)	
(paper)	(acc.)	(outside)	(throw.pst)	([Someone] threw the paper outside)
2:[tag="NN"]	[tag="PSP\:को"]	[tag=”NN”]	1:[tag="VM"]	------- (2)

Figure 12. A sample of similar rules for different dependency relations

In (1), the ergative marker indicates that the noun (NN) is the *doer* of the verb (VM). In (2), the accusative marker indicates that the noun (NN) is the *object* of the verb (VM). Also, (2) is not a complete sentence – the *doer* has not been mentioned, and only the part of the sentence that the rule is applied to is shown.

If in the rules, the **PSP** POS tags didn't contain the lexical features, both the rules would have been the same, and hence both the rules have been applied on both the sentences, making the word sketches erroneous.

By lexicalizing the **PSP** POS tag, the rule(s) formed are now less ambiguous, and more accurate.

After extracting all the dependency rules, we apply each rule on the annotated Treebank (HDT), and compute its precision. For example, if the rule "**k7 (location)** - 2:[tag="NN"] [tag="PSP\:में"] [tag="NN"] 1:[tag="VM"]"
is applied on the HDT, we get all the [2:NN, 1:VM] pairs where the rule holds, say **N** pairs. Out of these **N** pairs, if **M** of them are seen correctly with k7 (place) relation in the training data then the precision of the rule is $\frac{M}{N}$.

Conditions that need to be satisfied for a rule to be included in the sketch grammar:

- The rule must have a precision of at least 70%. A higher cut-off gives us rules with better precision, but less recall, and it is the reverse for lower cut-off limits.

- The frequency of the tag sequence (N) must be greater than 4, to ensure some amount of statistical significance of the rules.

The context size – the maximum allowed length of the tag sequence (rule), is set to 7 to limit the number of rules generated.

4 Error Analysis

The word sketches may not always be accurate due to the ambiguous nature of rules, and POS tagging errors. For example, when a rule such as
1:[tag=”NN”] [tag=”PSP: ने”] 2:[tag=”VM”]
is applied on sentences which have nouns ending with the honorific जी in the data, जी is likely to be a collocate of the words. This is an error due to the POS tagger that we use to tag the HindiWaC. The tagger tags the honorific जी as **NN** and not **RP**. These word sketches can be improved further by improving the POS tagger, or the grammar, for example by involving local word group information. The other related errors are due to **UNK** POS tag which is generally assigned to the words that are unknown to the tagger.

As the length of a rule increases so does its sparsity, i.e. the number of sentences on which the rule can be applied since all the POS tags in the rule have to occur in that particular order. Hindi being a free word order language the possibility of all the POS tags occurring in that order is even less.

5 Conclusion and Future Work

Corpora are playing an increasing role in all kinds of language research as well as in language learning, lexicography, translation, literary studies and discourse analysis. The requirements are, firstly, a suitable corpus, and secondly, a corpus query tool. We have presented HindiWaC, a large corpus of Hindi, which has been prepared for use in the Sketch Engine. We have described how we used Hindi Dependency Treebank to develop the grammar underlying the word sketches. And we have shown the core features of the Sketch Engine, as applied to Hindi.

Our current grammar is prone to data sparseness as the length of rule increases. A future direction of this work could be on building compact grammars using regular expressions.

Additionally, one could also explore the usefulness of morphological features in grammar rules to make them semantically accurate.

Reference

Adam Kilgarriff, Pavel Rychly, Pavel Smrz and David Tugwell. *The Sketch Engine*. Proceedings of EURALEX 2004.

R. Begum, S. Husain, A. Dhwaj, D. Sharma, L. Bai and R. Sangal. *Dependency annotation scheme for Indian languages*. Proceedings of International Joint Conference on Natural Language Processing 2008

Adam Kilgarriff, Siva Reddy, Jan Pomikálek and Avinesh PVS. *A Corpus Factory for many languages*. In *LREC*. 2010

Jan Pomikálek. *Removing Boilerplate and Duplicate Content from Web Corpora*. 2011

Siva Reddy and Serge Sharoff. *Cross Language POS Taggers (and other Tools) for Indian Languages: An Experiment with Kannada using Telugu Resources*. Proceedings of IJCNLP workshop on Cross Lingual Information Access 2011.

Adam Kilgarriff and Michael Rundell. *Lexical Profilng Software and its Lexicographic Applications – a Case Study*. In A. Braasch et al. (eds.) EURALEX Proceedings 2002 807-818.

A. Bharati, R. Sangal, D. Sharma and L. Bai. "Anncorra: Annotating corpora guidelines for pos and chunk annotation for Indian languages." LTRC-TR31 2006

D. Sharma, P. Mannem, Joseph van Genabith, Sobha Lalitha Devi, Radhika Mamidi and Ranjani Parthasarathi. *Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012)*. Proceedings of International Conference on Computational Linguistics (COLING) 2012

Kilgarriff, Adam, et al. "Itri-04-08 the sketch engine." *Information Technology*105 (2004): 116.

Pearce, Michael. "Investigating the collocational behaviour of man and woman in the BNC using Sketch Engine 1." *Corpora* 3.1 (2008): 1-29.