The lists:

* Representative

The lists must be representative of how people talk on Twitter about each issue. This means that they must not be skewed towards any particular sub-topic, and must be broad enough to cover the general arena.

* Discriminatory

The list words must be narrow enough to discriminate tweets with some certainty. Some words in themselves might not indicate that the tweet is relevant to the issue. For example, #activism may include tweets about other campaigns. The identifying process must take this into account.

* Equally rigorous in Hindi/English for a given topic:

It is important that the four lists (Hindi, English, transliterated Hindi into Roman script, transliterated English in Devanagri script) are **equally** comprehensive. Since we are studying relative uses of Hindi and English, it will skew the results if, say, the English list is weaker, i.e. less catching, than the Hindi one.

* Equally rigorous across topics:

Similar to the above point – we are making a comparative study, and so require that the caste-concerns lists and the gender-concerns lists are equally catching to avoid bias in the results as much as possible.

There are two categories of words:

Primary list: words/phrases that, standalone, can be assumed to indicate that the tweet in which they appear concerns the particular discussion. For example, #dalit, #feminism, #genderequality.

Secondary list: words which must appear with at least one supporting word to be indicative of a particular discussion. For example, ‘domestic’ and ‘violence’ must appear together in a tweet for it to indicate a discussion on women’s rights.[[1]](#footnote-1)

How do we know that the lists are rigorous?

or rather,

How do we know that the relative rigour of the Hindi and English lists is the same?

How sure can we be that the tweet identified concerns the target topic?

What about out-of-topic tweets? Is it okay to include them in our analysis?

These lists are not comprehensive, of course, as identifiers for discussions on either of the topics.

Keyword Extraction:

Essentially, we are claiming that a given list of words comprises a set of keywords for a given discussion. Based on this assumption, we are statistically analyzing the incidence of these keywords in Hindi and English. A more data-based approach to the same problem would be the creation of the lists by a keyword extraction model that can process code-mixed data and create a Hindi-English combined list of keywords. This is as yet unavailable to us; we have done this task manually.

This comes with its risks. There are certainly keywords that we are missing. We may possibly be missing different numbers of these in Hindi and English which will bias the results. For example, in a tweet:

Brahman Baniya media was and is always against the rights of [#**Bahujan**](https://twitter.com/hashtag/Bahujan?src=hash) Rather acting as fourth pillar of democracy,it acted as slave of [#BrahmincalPatriarchy](https://twitter.com/hashtag/BrahmincalPatriarchy?src=hash) [@htTweets](https://twitter.com/htTweets) atrocious manuwadi mindset

Iske khilaaph avaaz uthaiye morcha lagaiyye

Suppose we had not identified ‘morcha’ as a Hindi-list keyword, we would overlook the key-contribution of the last Hindi sentence to the overall tweet (although it would appear, of course, in run length analysis).

Stemming, case normalization, punctuation:

We will be using the nltk package for this.

<https://www.datacamp.com/community/tutorials/stemming-lemmatization-python>

<https://machinelearningmastery.com/clean-text-machine-learning-python/>

Process:

Manually identify conversations. Search Hindi-English terms in conjunction: e.g. ‘aurat #honour killing’.

**Q:** How rigorous or biased is the corpus we get?

Now identify

1. Simple percentages
2. Significant keywords: English list, Hindi list, for each topic. Compare for caste vs. feminism. **Q:** How equivalent are these two lists?
3. Now nuance this. Pick up specific terms: honour killing as it appears in hindi text, and liberation as it appears in hindi text. The first is culturally entrenched, will appear more. Similarly for caste.

18/3

429 errors: Too many requests per minute. Only collecting about 5 tweets for one word combination (a pretty catching one at that).

No handlers found for logger tweepy.binder

Reading and searching Hindi terms: UTF-18 encoding not working –

Non-ASCII character '\xe0' in file scraper.py on line 21, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details

The problem, still, is just collecting the data.

20/3

Suddenly, it is working. For ‘dalit’. (single term search)

Problems:

1. RTs (ignore?((Nobody will know)))
2. Repetition across search terms (Soln: search combinations, reducing the probability of repeated terms, since 4 terms are not going to appear)
3. Searching Hindi keywords (AASCII)

Not all combinations are will yield relevant results.

To do:

**Create these combinations.**

1. Trigger/ non-trigger.

Get ER. Convert to ED. Translate non-nouns to HD. Supplement and finalize HD. Convert to HR.

22/3

utf-8 encoding actually works. Why am I so surprised when technology doesn’t fuck up hmm I wonder.

We have 24000 search terms.

Plan:

1. Edit generate\_searchterms to remove repeat terms -\_-
2. Create for fem
3. Generate F\_Search Terms
4. Mine data (overnight?)

Problems currently

1. Can’t think of problems

Something must be wrong.

27/3

Tweets: 70000

1. RT issues
2. Mining efficiently: running each combination is taking too long.   
   1. Reduce each list proportionately (this is out)  
   2. Run each search for fewer items (adopting this one)
3. Cleaning
4. Langid.py/ throw it at MSR (mono must be emailed)

Preetha:

1. Figure out RT
2. Choose randomly from given data.
3. Cleaning data – if this is necessary

Niyati

1. Analysis model
2. Blog post

In general:

1. Report

28/3

Work done:

RT’s removed  
Randomly reduced the data.

LID:

Data has to be stripped of Devanagari, emoticons, URLS, hashtags, etc.

29/3

Analysis Model

Given: Dalit politics corpus, feminism corpus, in intermixed Devanagari and Roman script.

We can perform the following analyses:-

1. Analysis of the occurrence of a list of ideological words, appearing in their Hindi and English equivalents, to determine the trends in ideological discussion.
2. Sentence-based analysis of relative Hindi presence.
3. Word-based analysis of relative Hindi presence
4. Identify tweets with CS.
5. Identify run length of CS when it exists: estimate whether it is phrasal or at the sentence level.

Methodologies:

1. This is pretty straightforword. Prepare the list, run a counting search on non-LID’d data.

LID:

For the next four, we will LID the corpora. We will also make certain assumptions(?):

Assumption 1 – There are very few English sentences written entirely in the Devanagri script, (although we might have long segments of Roman transliterated Hindi). This seems true from observation in the corpora – it is also fairly intuitive, since the English – Hindi Roman – Devanagri baseline for Twitter is 70-15-15; it is probably rare that people typing in the Devanagri script are expressing themselves in English. In any case, we can observe our own corpus to validate this assumption.

Assumption 2 – There is very little *script*-mixing in one sentence. Again, this is true from observation. People do insert English acronyms (SC, BJP, etc.) into their Devanagri tweets, but they do not switch scripts halfway through a sentence, or for particular phrases.

(Possibility: We can remove the sentence which have script mixing. Will this matter?)

The necessity for the above two assumptions arise from the limitation that we do not have an en-hi LID software for Devanagri script. However, both the assumptions are in fact reasonable, and will probably result in only marginal error.

**In general, we assume that sentences are either in Roman or in Devanagri script, with perhaps some lexical level insertions in the other script, which we can ignore. Furthermore, a sentence in Devanagri script is a Hindi sentence, although it may contain some transliterated English words.**

Now, we separate the Devanagari sentences from the corpus, and count them. These are our Hindi tweet-sentences. They contain some CS, of course, but we do not at present have the resources or the scope to measure this.

(One method: Run each word by a Hindi dictionary. If no match is found, it must be an English word. This approach has several holes – abbreviations, acronyms, slang, etc. will not show up in a dictionary.)

We LID the Roman script corpus.

Assumption 3 – The language of a sentence is usuallythat in which the larger number of words is present. This tends to be true in general, but not, of course, always – e.g. in the sentence “She told me to come but I said that **Hindi phrase HP**” the sentence is English, no matter how long HP is. However, in most cases, this assumption holds.

1. Given the LID’d corpus, we do a hi-en comparison for each sentence, and assume its language is that which has greater word presence. We append Hindi count (HR\_count) to our previous Hindi Devanagari HD\_count. This gives us a sentence-based analysis, i.e. analyzing at the lowest unit of a conversation, and ignoring CS for the moment. The reason we are choosing the sentence as a unit and not a tweet is because a multiline tweet may contain two lines, say, in different languages, in which case the dominant language cannot be considered at that containing the most words (in fact, we may say that there is no such language). That is, we are trying to analyse, simplistically, the language in which this conversation is occurring at sentence level.
2. This analysis can only be performed on the LID corpus, since we cannot identify English Devanagri.
3. Once again, we cannot determine this heuristic exactly, for the same reason. However, we can calculate two separate figures, to provide a rough idea of the number of tweets with CS:  
   1. Firstly, while separating the Devanagri, we count the number of tweets which contain inter-sentential script mixing, i.e. sentences in different scripts (Note that this does not counter Assumption 2, where we are talking about mid-sentence script-switching. People do, in fact, switch scripts between lines in multiline tweets). We know that these tweets definitely contain CS (given Assumption 1): this is ScriptMix\_count.  
   2. After LIDing, count the tweets with English and Hindi mixed: RomanCS\_count.  
   Of course, there is a factor of double counting here: given a multiline tweet which contains a Devanagari sentence *and* Hindi-English mixing in Roman script lines, we will count it twice. This is our error margin.   
   We do obtain, in any case, that the number of code-switched tweets is in the range (max(ScriptMix\_count, RomanCS\_count) , (ScriptMix\_count + RomanCS\_count))
4. Given a line, we can find the run length of CS, if present. We have labelled the language of the line in (2). Typically, a run-length of 3 or more in the opposite language indicates phrasal CS. We can count, in fact, EP type lines and HP type lines: this figure gives us greater indication than (4) as to how fluid the lingo is between Hindi and English i.e. how comfortable are people switching back and forth mid-sentence in this discourse, as compared to between sentences. Of course, this count is a subset of (4).

Questions:

1. LID: what about words like Dalit?
2. Assumptions
3. Work with Hindi dictionaries for (3)?

Find switchpoints.

1. mainly, although it may also be discussing domestic violence against men, of course. Our approach to the incidence of the latter, that would result in an ‘out-of-topic’ tweet, is mentioned below [↑](#footnote-ref-1)