NLP Term Project Report

Group 6, Project 16: Code-mixing patterns in celebrities

Description

In this project, we want to recognise code-borrowing instances on social media, Twitter. Primarily, we want to understand how a new foreign word introduced by a celebrity is used by their followers, and how they might be incorporated in the native language over a period of time, and understand underlying dynamics of borrowing flow
Also, we also want to understand sense deviation of English words which are used in Hindi context on social media. We aim to inspect which senses are used more probably than others for an English noun.

Objective

We aim to analyse English nouns which celebrities use, and how the flow of such code-borrowed words occurs in Hindi context. We shall do this by extracting UUR, UTR, UPR ratios for each word (defined later on).
Further, for each English word used in Hindi context, we identify which senses best describe the English word. We also want to check how similar English nouns and English nouns in Hindi contexts are. The following section describes the procedure followed.

Experiments and Procedure

TASK-1: Formatting the Cleaned Data

Input Format

The cleaned data contains two lines for each tweet in the raw data. The first line contains the tweet-id and the tweet. The second line contains the word level tagging of the tweet based on the language(EN or HI).

Language tagging is done by using the Microsoft language tagger.

Output Format

The data is formatted and stored in the form of dataframe in a pickle file. If needed, this dataframe can be simply converted into json format using an inbuilt function.

The dataframe contains the following columns:

- tweetid : Contains the id of the tweet
- isCeleb : Contains 0 or 1 to indicate whether the person is a celebrity or not.
- Tweet : Contains the original tweet.
- Tweet-tag : Contains the tweet-tag of the tweet.
- Word-level : Contains the word-tag and context-tag for each of the words in the tweet

The "Word-level" is in turn a dataframe with words of the tweet as indexes and 'word-tag' and 'context-tag' as columns. The value of ‘isCeleb’ is decided based on whether the tweet is a normal tweet or a retweet.

Tweet tag

The tweets in the data set are classified into six different categories.

- En: Tweet contains 90% or more English words.
- Hi: Tweet contains 90% or more Hindi words.
- CMH: Code-mixed tweet but majority(>50%) words in Hindi.
- CME: Code-mixed tweet but majority(>50%) words in English.
- CMEQ: English and Hindi words almost same number of occurrences.
- CS: (Code-switched) Trail of En / Hi words followed by trail of Hi / En words.
- OTHER : Tweet contains only mentions or URLS.

The tweet-tag for a tweet is assigned based on the word-level tags in the tweet.

Code-Mixed and Code-Switched Classification

The word tags of the tweet are taken into an array or a list. Then, the number of positions at which the word-tag changes is calculated.

- If the number of changes is one, then the tweet is code-switched tweet.
- If the number of changes is greater than one, then the tweet is code-mixed tweet.

Context tag

The context tag for each of the words in a tweet is assigned based on the tweet-tag of the tweet.

- If the tweet-tag is either 'EN' or 'CME', then the context-tag is 'EN' for all the words in the tweet.
- If the tweet-tag is either 'HI' or 'CMH', then the context-tag is 'HI' for all the words in the tweet.
- If the tweet tag is 'CMEQ', then the context tag is 'O' for all the words in the tweet.
- If the tweet tag is 'CS', then the context-tag is same as the word-tag.

Named Entity Recognition

The named entities in a tweet can be extracted by tagging parts-of-speech using the Natural Language Toolkit.

The word-tags of these words are changed from 'EN' or 'HI' to 'NE' in the dataframe.

TASK-2: Dataset Analysis

A word in some language , is said to be borrowed by Language2 , if it is used as a native language word, instead of using the native language meaning . Ex : College is used as a native hindi word, instead of विद्यालय . Hence, it is a borrowed word from English.

We are interested in exploiting different measures to efficiently predict, if a word is likely to be borrowed in future. We use Hindi as L1 and English as L2. We analyse the patterns of borrowed words from english to hindi. We observe the patterns between the tweets of celebrities and their followers.

We use three different metrics to analyse the words in their tweet. The different tags that a word can have are: L1, L2, NE (Named Entity) and Others.

Based on the word level tag, we also create a tweet level tag as follows:

1. L1: Almost every word (> 90%) in the tweet is tagged as L1.
2. L2: Almost every word (> 90%) in the tweet is tagged as L2.
3. CML1: Code-mixed tweet but majority(i.e., > 50%) of the words are tagged as L1.
4. CML2: Code-mixed tweet but majority(i.e., > 50%) of the words are tagged as L2.
5. CMEQ: Code-mixed tweet having very similar number of words tagged as L1 and L2 respectively.
6. Code Switched: There is a trail of L1 words followed by a trail of L2 words or vice versa.

Using the above classification, we define the following metrics:

- ***Unique User Ratio (UUR)*** –The Unique User Ratio for word usage across languages is defined as follows:

$$UUR(w) = \frac{U_{L_1} + U_{CML_1}}{U_{L_2}}$$

- (1)

where ***UL1 (UL2, UCML1)*** is the number of unique users who have used the word ***w*** in a ***L1 (L2, CML1)*** tweet at least once.

- ***Unique Tweet Ratio (UTR)*** – The Unique Tweet Ratio for word usage across languages is defined as follows:

$$UTR(w) = \frac{T_{L_1} + T_{CML_1}}{T_{L_2}}$$

- (2)

Where ***TL1 (TL2, TCML1)*** is the total number of ***L1 (L2, CML1***) tweets which contain the word ***w***.

- ***Unique Phrase Ratio(UPR)*** – The Unique Phrase Ratio for word usage across languages is defined as follows:

$$UPR(w) = PL1 / PL2 \quad \text{- (3)}$$

where ***PLx*** is the number of ***L1/L2*** phrases which contain the word w.

Note that unlike the definitions of UUR and UTR that exploit the word level language tags, the definition of UPR exploits the phrase level language tags. We then calculated Jaccard Index and Spearman Correlation between

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}. \quad \text{- (3)}$$

We calculated Jaccard Index between celebrities and followers words (total). We ordered sets of (word,metric) pairs of size 50, 100, 200, 300, 400 in increasing order of their metric value. e then Calculated Spearman correlation for metrics of followers and celebrities for all the sets .

<u>TASK-3</u>: Sense Deviation

Extracting English nouns

For this, we shall first distinguish English nouns in English context and English nouns in Hindi context. We append an '$' to every english word in Hindi context.
So, “***Aaj mein film dekhne gya”*** becomes “***Aaj mein film$ dekhne gya”***.

Representing nouns in Social media

Once we separate English nouns in English and Hindi contexts, we need to represent words like “***film”*** and “***film$”*** differently according to their occurrences in Social media.

We create 2 word2vec models, where one has words with minimum occurrences as 5, while the second model considers all words, and uses multiple workers.

Obtaining possible senses for English nouns

We identified 6846 English nouns for which senses were to be identified. For a sample word, say ***film***, we identify synonyms for the word using wordnet. Now, we shall translate every synonym to Hindi using google translate programmatically. All these translated Hindi words are then assigned as possible Hindi senses for the word ***film***.

Now, to test our sense analysis, we shall separate a target test set of 57 words ***('thing', 'way', 'woman', 'press', 'wrong', 'well','matter', 'reason', 'question', 'guy', 'moment', 'week', 'luck', 'president', 'body', 'job', 'car', 'god', 'gift', 'status', 'university', 'lyrics', 'road', 'politics', 'parliament', 'review', 'scene', 'seat', 'film', 'degree','people', 'play', 'house', 'service', 'rest', 'boy', 'month', 'money', 'cool', 'development', 'group', 'friend', 'day', 'performance', 'school', 'blue', 'room', 'interview', 'share', 'request','traffic', 'college', 'star', 'class', 'superstar', 'petrol', 'uncle').***

These words were found using the work from All that is English may be Hindi: Enhancing language identification through automatic ranking of likeliness of word borrowing in social media.

The above words were found comparing the frequency of an English noun in English newspaper corpus and Hindi newspaper corpus. For each word, we identify the ratio

$$log\,[F(English,\ word)\ \div\ F(Hindi,\ word)]$$

Based on this ratio, we chose 30 words with extremely high and extremely low ratio, and 27 words with moderate ratio. These words were excluded from training set.

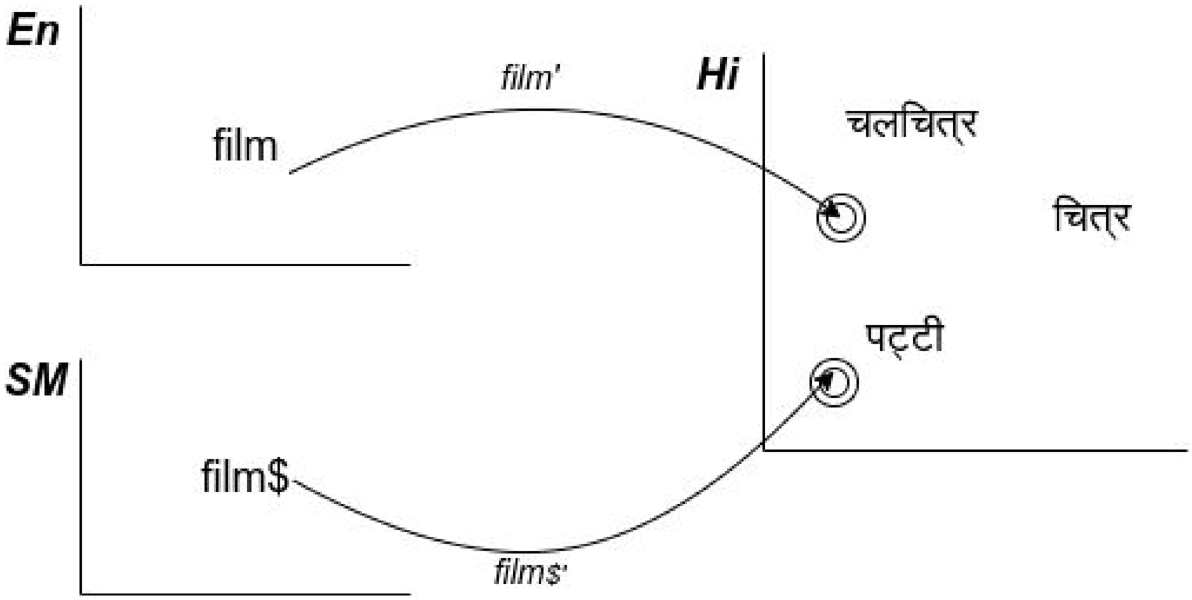
Visualising social media vector

Word2Vec is building word projections (embeddings) in a latent space of N dimensions, (N being the size of the word vectors obtained). We obtained coordinates of all words in our social media dataset; in 300 dimensions. We used Dimensionality reduction (TSNE) to visualise vectors in 2 dimensions. These points were then visualised using matplotlib.
We made visualisations for Naive-sm, hindi word2vec and complex-sm points .we observe that group of words forming clusters are closely related such as (school and college). We also extracted top 500 points , in increasing order of their pairwise distance and visualised them .

Evaluating possible used senses in social media

For each word ***film$***, we will evaluate what possible senses this word appears in our tweet dataset. This involves manual tagging for all words. Two annotators tagged all 2238 English nouns in Hindi contexts for possible senses. We found the inter-annotator agreement (Cohen's kappa value) as 0.633. We also analysed the number of senses each noun was assigned.

Transforming vectors to Hindi vectorspace



To properly compare words like ***film$*** in social media, and ***film*** in conventional English, we shall be transforming vectors in social-media vectorspace and English vectorspace to Hindi vectorspace.

$$\min_{W} \sum_{i=1}^{n} ||Wx_i - z_i||^2$$

Using the method described in Hamilton's (2016): Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change, we use procrustes method for minimising the aforementioned error. X_i refers to English / social media vector and Z_i is Hindi vector, where W is the transformation matrix.

Result Analysis

In some cases, we obtained a negative cosine similarity, which is possible in the method we used to generate word embeddings model. Refer this stackoverflow answer.

First we shall transform social media vector for ***film$*** and english vector for ***film***. We are trying to analyse 3 kinds of results.

Sense similarity

For an English noun E which appears in Hindi context (E$), we consider the possible senses (H1, H2, H3, ..). For every possible sense, we consider the cosine similarity of (E, H1) and (E$, H1). Using this, we were able to find if, for a certain sense, it is closer to E or E$.

For a sense Hx, if cosine(Hx, E$) > cosine(Hx, E), it means when an English word appears in Hindi context, it is more likely to mean in the sense of Hx.

Vector similarity

Now, for each English noun E which appears in Hindi context (E$), we consider the similarity of E and E$. If cosine(E, E$) is high, it means that the senses of E and E$ are closer to each other.

This is possible when E does not have multiple senses. Example, ***movie*** and ***movie$*** will have higher cosine similarity than ***film*** and ***film$***, because movie has very few possible senses, however film can mean cinema, strip, photographic film, etc.

Results

TASK-1: Formatting

After tagging 30 lakh tweets, we obtained the following statistics:
We obtained 7, 70, 234 tweets from celebrities and 13, 67, 490 tweets from followers.

Tweet-tag	Occurrences (Celeb)		Occurrences (Followers)	
English	452207	58.72 %	182794	13.37%

Hindi	7514	0.97%	23667	1.731%
Code-mix En	215390	27.97%	75886	5.55%
Code-mix Hi	16608	2.15%	23999	1.75%
Code-mix Equal	6791	0.88%	8187	0.59%
Code-switched	42905	5.57%	47970	3.51%
Other	28595	3.71%	1004705	73.48%

After removing stop words, we extracted 203 unique nouns from celebrities' tweets and 1804 from followers' tweets.

TASK-2: Dataset Analysis

We ranked words according to UTR, UUR and UPR values, which are tabulated as follows:

	Celebrities		Followers	
	Top 100th value	Top value	Top 100th value	Top value
UUR	0.04	11	1	34
UTR	0.034	13	1	36.45
UPR	0.023	1	0.555	3

The top words are available on the source code. Some top words were 'memo', 'powder', 'spelling', 'muscle', 'leather', 'keyboard', etc.

Word-size	Jaccard	spearman-utr	spearman-upr
100 (initial)	0.074	0.252	0.31
50	0.102	0.62	-1
100	0.102	0.804	0.247

200	0.102	0.56	0.417
300	0.102	0.487	0.444
400	0.102	0.496	0.439
500	0.102	0.411	0.388
1000	0.102	0.351	0.437

As we increase the size of our top list, spearman-utr decreases; while spearman-upr increases initially and then swings within some range.

TASK-3: Sense Deviation

Extracting English nouns and possible senses

We identified a total of 20, 899 tweets, which contain at least one English noun in Hindi context. A total of 2238 English nouns were used in Hindi context. However, senses were identified for 1844 nouns, some of the other 394 words which are Hindi words, but are also present in English nouns, examples of which are “bus”, “mat”.

Of the 1844 nouns, 1456 words had only a singular sense, while the rest 388 have multiple senses. Of the 1456 words with a singular sense, 154 words have transliterated word as the sense, like desk: ['डेस्क'] . In words with multiple identified senses, 71 of 388 words with multiple senses have a transliterated word as the sense, for example, “सर्विस” as the best sense of “service”.

We analysed why the transliterated word made for the most probable sense. For those 225 nouns, we found some of them were proper nouns, “England”, “Indonesia”, etc. Technical words were borrowed as it is, without change, for example “internet”, “site”, “keyboard”, “calculation”. However, we also found instances of words borrowed from Hindi to English, like “yoga” and “dacoity”.

The organised results can be found on this Google sheet.

Sense similarity

For every English noun (E) in Hindi context (E$), we calculate the similarity of (E, Hi_x) and (E$, Hi_x), and then compute their difference; where Hix is one of the senses of possible senses (Hi_1, Hi_2, $Hi_{3,}$, ,....)

We define cosine difference as = cos(E, Hi_x) - cos(E$, Hi_x)

We observed that if cosine difference is positive, the sense Hi_x is closer to the English word in conventional English than in Hindi context on Twitter. However, if cosine difference is negative, the sense Hi_x more strongly represents E$ than E.

We found words with a single meaning were more likely to have a negative cosine difference, and nouns with multiple senses are more prone to have a positive cosine difference.

Thus, the cosine similarity of “movie” and “movie$” will be negative, since “movie” is used only in one sense. The cosine similarity of “film” and “film$” will be positive, since film has multiple meanings in English.

English word	Social Media	Hindi sense	cosine difference	Hindi sense	cosine difference
Star	Star$	तारा	0.0982	सुपरस्टार	-0.180
Scene	Scene$	स्थल	0.275	शॉट	-0.311
well	well$	अच्छा	-0.104	काफी	0.0259
press	press$	दबाएँ	-0.102		
blue	blue$	निराशाजनक	-0.133	नीला	0.38

We see for star means सुपरस्टार when used in Hindi context, similarly for (scene, शॉट) and (blue, निराशाजनक) .

Vector similarity

For an English noun (E) in hindi context (E$), we compute the cosine similarity of the transformed vectors. In some cases, we obtained a negative cosine similarity, which is possible in the method we used to generate word embeddings model. Refer this stackoverflow answer.

We found English nouns(E) in hindi context (E$). used in singular sense /

were used in same sense in Hindi and English contexts.

English	SM	Cosine similarity
god	god$	0.2421904787
blue	blue$	0.127730732
president	president$	0.1222979614
woman	woman$	-0.03398525207
job	job$	-0.1183152846
play	play$	-0.1574467486

We found words with singular meaning like “god”, “president” had high cosine similarity, and words with multiple meanings like “job”, “play” tend to have negative cosine similarity. However, we also found for “woman”, which has a singular meaning has negative similarity.

Conclusion

We have implemented transformation matrix using Hamilton(2016) with procrustes method, without using any external library’s function. We detected that certain senses are closer to an English noun in Hindi context as compared to the same word in English context. However, we noted we need more data to fully understand the trends and sense distribution for English words in social media.

Group members

Kaustubh Hiware, 14CS30011
Surya M., 14CS30017
T. Karthik , 14CS10049
G Prithvi Raj Reddy, 14CS10016
Kiran Sing Sastry.G. 14CS10018

The code for this project is available on GitHub.
The slides are available on Slides.