Annotation Efficient Language Identification from Weak Labels

Shriphani Palakodety*
Onai
spalakod@onai.com

Ashiqur R. KhudaBukhsh*
Carnegie Mellon University
akhudabu@cs.cmu.edu

Abstract

India is home to several languages with more than 30m speakers. These languages exhibit significant presence on social media platforms. However, several of these widely-used languages are under-addressed by current Natural Language Processing (NLP) models and resources. User generated social media content in these languages is also typically authored in the Roman script as opposed to the traditional native script further contributing to resource scarcity. In this paper, we leverage a minimally supervised NLP technique to obtain weak language labels from a large-scale Indian social media corpus leading to a robust and annotation-efficient language-identification technique spanning nine Romanized Indian languages. In fast-spreading pandemic situations such as the current COVID-19 situation, information processing objectives might be heavily tilted towards under-served languages in densely populated regions. We release our models to facilitate downstream analyses in these low-resource languages[1]. Experiments across multiple social media corpora demonstrate the model's robustness and provide several interesting insights on Indian language usage patterns on social media. We release an annotated data set of 1,000 comments in ten Romanized languages as a social media evaluation benchmark[1].

1 Introduction

Much of the current NLP research focuses on a handful of world languages (e.g., English, French, Spanish etc.). They enjoy substantially larger computational linguistic resources as compared to their low-resource counterparts (e.g., Bengali, Odia etc.). However, in the midst of global-scale events like the ongoing COVID-19 pandemic, demand for linguistic resources might get recalibrated; information processing objectives might be heavily tilted towards under-served languages that are prevalent in many densely populated regions.

In this paper, we focus on *language identification* in noisy, social media settings - a basic and highly critical linguistic resource prerequisite for downstream analysis in a multilingual environment. Our solution extends support for nine major Indian languages (see, Table 1) spanning the native tongues of 85% of India's population (Census, 2011). These under-resourced languages are heavily used in several densely populated travel hubs and on social media. User generated web content in these languages is typically authored in the Roman script as opposed to the traditional native script leading to scarcer linguistic resources (Virga and Khudanpur, 2003; Choudhury et al., 2010; Gella et al., 2014; Barman et al., 2014; Palakodety et al., 2020a). Existing large-scale language identification tools prioritize the languages' native scripts (e.g., (FastText; Google)) over the Romanized variants. Our solution focuses on the these Romanized variants and is integrated with a widely used existing language-identification system (FastText) supporting 355 languages. We **release our open-source language identification system**[1]. to facilitate Indian social media analysis.

Annotator availability is a major concern that may constrain data acquisition efforts in low resource settings (Joshi et al., 2019). Our proposed solution is extremely **annotation efficient**; it utilizes a recent result (Palakodety et al., 2020a) to automatically group a multilingual corpus into largely monolingual clusters that can be extracted with minimal supervision. Using a mere 260 annotated short documents (YouTube video comments), we

* Shriphani Palakodety and Ashiqur R. KhudaBukhsh are equal-contribution first authors. Ashiqur R. KhudaBukhsh is the corresponding author.

[1] Resources are available at: https://www.cs.cmu.edu/~akhudabu/IndicLanguage.html.

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 181–192
Online, Nov 19, 2020.

assign *weak labels* to a data set of 2.8 million comments spanning the aforementioned languages. Our model performs favorably when compared against an existing commercial solution.

While census data and surveys can provide useful information about linguistic diversity and spread, analyses of user-generated multi-lingual corpora can complement these surveys with additional useful insights of their own. We conduct a focused analysis to explore if the (estimated) distribution of web-usage of Hindi across different Indian states aligns with common knowledge. Our analysis indicates that Hindi's web-presence is considerably higher in a cluster of North Indian states referred to as the Hindi belt (Jaffrelot, 2000) as compared to the South Indian states. We further analyze similar research questions concerning the relative usage of the Roman script and the native script for Hindi. We finally conclude with a small exploratory study on our method's effectiveness in detecting languages with trace presence in multiple corpora and outline some of the possible utilities.

Contributions: Our main contributions of the paper are the following:

- ***Resource***: We release an important linguistic resource to detect nine heavily-spoken Indic languages expressed in Roman script.
- ***Method***: We propose an annotation efficient method to construct this language identifier and demonstrate extensibility.
- ***Linguistic***: We conduct a web-scale analysis of Hindi usage shedding light on multilinguality, geographic spread, and usage patterns.
- ***Social***: We outline how our tool can detect trace presence of other languages that can aid in constructing data sets for humanitarian challenges.

2 Data Set: YouTube Video Comments

In order to construct our language identification system, we would require a web-scale Indian social media data set that (i) has considerable presence of the nine languages we are interested in, and (ii) captures a representative fraction of the Indian web users. To achieve this two-fold goal, we consider a data set introduced in Palakodety et al. (2020b) to analyze the 2019 Indian General Election. The data set consists of comments on YouTube videos hosted by popular news outlets in India. Overall, the corpus consists of 6,182,868 comments on 130,067 videos by 1,518,077 users posted in a 100 day period leading up to the 2019 Indian General Election.

Why YouTube? As of January 2020, YouTube is the second-most popular social media platform in the world drawing 2 billion active users (Statista, 2020). YouTube is the most popular social media platform in India with 265 million monthly active users (225 million on mobile), accounting for 80% of the population with internet access (HindustanTimes, 2019; YourStory, 2018). YouTube video comments have been used as data sources to analyze recent important events (Palakodety et al., 2020a,c; Sarkar et al., 2020; Cinelli et al., 2020).

Language	ISO code	First language speakers
Bengali	*bn*	8.03%
Gujarati	*gu*	4.58%
Hindi	*hi*	43.63%
Kannada	*kn*	3.61%
Marathi	*mr*	6.86%
Malayalam	*ml*	2.88%
Odiya	*or*	3.10%
Tamil	*ta*	5.70%
Te	*te*	6.70%

Table 1: List of languages we considered with their corresponding ISO 693-1 codes and first language speakers as percentage of Indian population. Data is collected from 2011 census (Census, 2011).

Why this data set? The data set considers two highly popular YouTube news channels for each of the 12 Indian states that contribute 20 or more seats in the lower house of the parliament. State boundaries in India were drawn along linguistic lines (Dewen, 2010). The dominant regional language in the Hindi belt (Jaffrelot, 2000) is Hindi, and the other states feature a unique dominant language written in either the Latin alphabet (in informal settings) or a native script. All the nine languages we focused on (listed in Table 1), are the dominant language in one or more of these 12 states. The regional news networks considered provide coverage in the dominant regional language. Hence, the data set exhibits strong presence of all the nine regional languages we are interested in. In addition to these 24 regional news channels, the data set considers YouTube channels for 14 highly popular national news outlets (listed in the Appendix). Overall, this implies 38 YouTube channels (24 regional, 14 national) with an average subscriber count of 3,338,628.

3 Related Work

Learning from weak labels: The role of unlabeled and weakly (or noisily) labeled data in supervised learning is a well-studied problem and has received sustained focus (Mitchell, 2004; Donmez et al., 2010), and annotation efficiency in low-resource settings is a well-established requirement (Joshi et al., 2019). Our work leverages $\hat{\mathcal{L}}_{polyglot}$ (Palakodety et al., 2020a), a recently-proposed method for noisy language identification that requires minimal supervision. We utilize it as a dependency to obtain *weak labels* and reduce annotation burden and construct a substantially more robust system.

Language identification: While language identification of well-formed text is a nearly-solved problem, the difficulty in identifying language in a noisy social media setting is well-established (Bergsma et al., 2012; Gella et al., 2014; Lui and Baldwin, 2014; Jaech et al., 2016; Jauhiainen et al., 2019). We see our work as a part of this continuing trend and as an important resource contribution to analyze Indian social media.

Romanized Indian Languages: In the context of processing Indian languages expressed on the web, challenges posed by the use of Roman script instead of the native script have been reported in several recent studies in the context of code-mixed English-Bengali (Chanda et al., 2016), and English-Hindi (Kumar et al., 2018) text. While addressing word level language identification, (Gella et al., 2014) reported that 90% of posts in Indian languages on Facebook are expressed in Roman script. Prevalence of Romanized Hindi has also been previously reported in (Barman et al., 2014). Our study takes previous findings one small step forward with a (noisy) geographical analysis of Hindi web usage.

Bridging the resource gap: We also see our work as a part of the ongoing effort in bridging the resource gap between Indian languages and world languages (Vyas et al., 2014; Vijayakrishna and Sobha, 2008; Kunchukuttan et al., 2014; Mohanty et al., 2017; Joshi et al., 2020).

4 Background

In this section, we summarize a few key NLP models and results critical to our methods.

Skip-gram embeddings: The Skip-gram model takes as input a word $w \in W$ (vocabulary), and predicts words $w_c \in W$ that are likely to occur in the context of w. The training objective (predicting an input word's context) is parameterized by real-valued word representations or embeddings (Mikolov et al., 2013). Bojanowski et al. (2017) introduced sub-word extensions to the Skip-gram model to learn robust word representations even in the presence of misspellings or spelling variations. Following (Palakodety et al., 2020a), we normalize and average a document's constituent word embeddings to yield the *document embedding*.

Monolingual cluster discovery: Palakodety et al. (2020a) introduced a minimal supervision language detection method using polyglot Skip-gram embeddings with sub-word information. These embeddings discover monolingual subsets (clusters) in a multilingual corpus which are subsequently retrieved using k-Means and a small sample per-cluster (10 documents) are annotated. We refer to this method as $\hat{\mathcal{L}}_{polyglot}$ and leverage it for constructing our data set with minimal annotation burden. For obvious reason, we do not compare $\hat{\mathcal{L}}_{polyglot}$ against our supervised solution that supports more than 300 languages. In Section 7.5, we demonstrate that our method detects languages with trace presence in a corpus ($< 1\%$), a known limitation of $\hat{\mathcal{L}}_{polyglot}$ (Palakodety et al., 2020a).

5 Method

Research question: *How to construct an annotation-efficient language identification method supporting a wide array of Indian languages?* Our method has two main components: (i) an annotation-efficient procedure to construct a substantial data set with weak labels, (ii) a supervised system trained on a data set comprising this corpus and an existing data set, $\mathcal{D}_{tatoeba}$ (Tatoeba, 2020), a well-known annotated data set supporting 355 languages (Tatoeba, 2020). For the construction of this data set, the election corpus (Palakodety et al., 2020b) is stripped of all comments containing any non English character. This maintains the focus on Romanized Indian languages with the native variants sourced from $\mathcal{D}_{tatoeba}$.

5.1 Assigning Weak Labels

Algorithm 1 outlines the steps in obtaining weak labels for nine Indian languages from multiple multilingual corpora and combining it with $\mathcal{D}_{tatoeba}$. Our training data set is denoted by $\mathcal{D} = \{d_i, \mathcal{L}(d_i)\}_{i=1}^{N_1} \cup \{d_i, \hat{\mathcal{L}}(d_i)\}_{i=1}^{N_2}$ where d_i is a document, $\mathcal{L}(.)$ returns a label annotated by a

Algorithm 1: $\mathcal{F}_{weakLabel}(\{\mathcal{D}_1, \ldots, \mathcal{D}_n\})$

Initialization:
$\mathcal{D} \leftarrow \mathcal{D}_{tatoeba}$
foreach $\mathcal{D}_i \in \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ **do**
  Run $\hat{\mathcal{L}}_{polyglot}$ on $\mathcal{D}_i$
  Obtain clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$ using k-means s.t. $|\mathcal{C}_1| \geq |\mathcal{C}_2| \ldots \geq |\mathcal{C}_K|$
  Identify $J \leq K$ dominant clusters
  for $(j = 1; j \leq J; j = j + 1)$ **do**
    Assign language to $\mathcal{C}_j$ (denoted by $\mathcal{L}(\mathcal{C}_j)$), (supplied by the annotator)
    Sample $\gamma|\mathcal{C}_j|$ comments from $\mathcal{C}_j$ ranked by proximity from cluster center, $0 < \gamma \leq 1$
    Add the sampled comments to $\mathcal{D}$ with *weak label* $\mathcal{L}(\mathcal{C}_j)$
  end
end
Output: Return $\mathcal{D}$

human, $\hat{\mathcal{L}}(.)$ returns a weak label obtained from $\hat{\mathcal{L}}_{polyglot}$.

$\mathcal{D}$ is initialized with an annotated corpus, $\mathcal{D}_{tatoeba}$ (Tatoeba, 2020), i.e., N_1 is the total number of samples present in $\mathcal{D}_{tatoeba}$. Next, using $\hat{\mathcal{L}}_{polyglot}$, our method obtains *weak labels* for N_2 documents from the election corpus and adds to $\mathcal{D}$.

Our method, $\mathcal{F}_{weakLabel(.)}$, takes an array of n multilingual corpora as inputs, runs $\hat{\mathcal{L}}_{polyglot}$ on each of them to obtain K language clusters, $\mathcal{C}_1, \ldots, \mathcal{C}_K$, such that $|\mathcal{C}_1| \geq |\mathcal{C}_2| \ldots \geq |\mathcal{C}_K|$. J largest of these clusters are selected and annotators assign a language $\mathcal{L}(\mathcal{C}_j)$, $1 \leq j \leq J$. For a given cluster $\mathcal{C}_j$, we obtain a set of pairs $\langle d, \hat{\mathcal{L}}(d)\rangle$ where $d \in \mathcal{C}_j$, and $\hat{\mathcal{L}}(d) = \mathcal{L}(\mathcal{C}_j)$, i.e., the weak label of the document is the cluster's language label. For each of these J clusters, the top γ fraction are chosen for inclusion into $\mathcal{D}$.

To summarize, for each multilingual corpus, $\hat{\mathcal{L}}_{polyglot}$ is used to obtain the top monolingual clusters, and a fraction of those are included with the cluster language label into the data set. Each cluster's language label is assigned by labeling 10 documents in the cluster and thus the vast majority of samples added to the data set is neither manually inspected nor labeled.

Recall that, each of the regional news outlets we considered presents news in one of the dominant regional languages. We group all comments obtained from the news outlets of one particular state as one distinct corpus - i.e. each $\mathcal{D}_i$ consists of comments posted in response to videos from a news outlet from a particular state. The choice of treating each individual state's corpus separately contributes further to annotation efficiency - knowing that a corpus is sourced from a region where a certain language is dominant allows us to select the appropriate annotators and reduces the annotation cost per document. We considered comments obtained from the 14 national outlets as a separate corpus. This led to 13 multilingual corpora (12 regional and 1 national). Hence, in our experiments, n, denoting the total number of corpora in Algorithm 1, was set to 13.

Parameter configuration: Our Algorithm has two configurable parameters: (1) j, the number of clusters selected per corpus for inclusion in the final data set, and (2) γ, the fraction of documents per cluster chosen for inclusion. We set j to 2. The choice of j was guided by the intuition that English is widely spoken in India and each state would have at least one dominant regional language. Our choice of γ was guided by an in-depth analysis of $\hat{\mathcal{L}}_{polyglot}$ in the context of code switching (KhudaBukhsh et al., 2020). The study revealed that documents closest to the cluster centers exhibit strong monolinguality and those farther from the centers can exhibit code-switching or may even be authored in languages with trace presence. In order to obtain high quality weak labels, we set γ to 0.75.

Annotation Efficiency: $\hat{\mathcal{L}}_{polyglot}$ requires 10 annotated samples to assign a language label to a cluster (Palakodety et al., 2020a). Our method requires $10jn$ annotated samples. Hence, our method required $10 \times 2 \times 13 = 260$ annotated comments to construct a corpus of 2,793,375 comments supporting nine Indian languages. This is combined with the $\mathcal{D}_{tatoeba}$ to yield $\mathcal{D}$ consisting of 11,042,839 documents.

True Label \ Predicted Label	*bn*	*en*	*gu*	*hi*	*kn*	*ml*	*mr*	*or*	*ta*	*te*	*ol*
bn	100	0	0	0	0	0	0	0	0	0	0
en	0	100	0	0	0	0	0	0	0	0	0
gu	0	0	100	0	0	0	0	0	0	0	0
hi	0	0	0	100	0	0	0	0	0	0	0
kn	0	0	0	0	99	0	0	0	0	0	1
ml	0	0	0	0	0	99	0	0	1	0	0
mr	0	0	0	0	0	0	100	0	0	0	0
or	0	0	1	2	0	0	0	96	1	0	0
ta	0	0	0	0	0	0	0	0	100	0	0
te	0	0	0	0	0	0	0	0	0	100	0
ol	0	0	0	0	0	0	0	0	0	0	0

Table 2: Confusion matrix of performance evaluation of $\mathcal{F}_{\textit{end-to-end}}$ on 1000 annotated comments. For a given language, better or equal performance than the baseline is highlighted with blue; *ol* denotes **o**ther **l**anguages.

True Label \ Predicted Label	*bn*	*en*	*gu*	*hi*	*kn*	*ml*	*mr*	*or*	*ta*	*te*	*ol*
bn	97	0	0	2	0	0	0	0	0	0	1
en	0	99	0	1	0	0	0	0	0	0	0
gu	0	0	92	4	0	0	4	0	0	0	0
hi	0	0	0	99	0	0	1	0	0	0	0
kn	1	1	2	0	86	5	0	0	0	3	2
ml	0	0	0	0	1	95	1	0	2	1	0
mr	0	0	1	0	0	0	99	0	0	0	0
or	20	0	13	18	2	3	16	0	0	4	24
ta	0	0	0	0	3	10	0	0	79	3	5
te	1	0	0	1	1	1	1	0	0	95	0
ol	0	0	0	0	0	0	0	0	0	0	0

Table 3: Confusion matrix of performance evaluation of `GoogleLandID` on 1000 annotated comments. For a given language, better or equal performance than $\mathcal{F}_{\textit{end-to-end}}$ is highlighted with blue; *ol* denotes **o**ther **l**anguages.

5.2 Learning with Weak Labels

Once $\mathcal{D}$ is obtained, we train a classifier that takes as input a document, and predicts the language label. We provide an end-to-end model operating directly on the text and producing a language label. The model utilizes a highly efficient text classification framework introduced in (Joulin et al., 2017). The framework introduces a variety of optimizations and is capable of classifying billions of documents in minutes without compromising on accuracy (implementation details are presented in the Appendix). We refer to this model as $\mathcal{F}_{\textit{end-to-end}}$. The model achieves comparable performance (test accuracy $> 98\%$) against a held out set that is not seen during any of the training phases.

6 Experimental Setup

Test set: We construct an Indian language test set consisting of 1,000 annotated YouTube comments (consensus labels by two proficient annotators per language) in 10 languages (Bengali, English, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Tamil, Telugu), and 100 documents in each language. These documents are randomly sampled from the output of $\mathcal{F}_{\textit{weakLabel}}$ and are never seen during our supervised training phase with weak labels. The average number of tokens in the comments is 22.6 ± 18.7. Please see Appendix for detailed statistics of our test data set.

Baseline: We consider a commercial solution (Google) that can detect over 100 languages including the 10 mentioned above (referred to as `GoogleLangId`) as our baseline. In addition to `GoogleLangId`, Palakodety et al. (2020a) compared against `FastTextLangID`. We avoid this comparison because we feel it is unfair to compare against `FastTextLangID` given that it does not support romanized Indic languages (achieves an overall accuracy 10% on our test set). We see our paper as a resource paper that makes a small step forward in addressing the lack of linguistic resources for Indian social media analysis.

Fairness: We first emphasize that the main purpose of comparing against `GoogleLangID` is **not to claim that our annotation-efficient solution is superior to `GoogleLangID` across the board**. Our goal is rather *to attract the research community's attention to our solution's effectiveness in this under-explored, specific domain of noisy social media texts generated in the Indian subcontinent*. A fairer performance comparison between the two methods would require the methods to be trained on identical data sets and comparable computation budget. Due to these varying levels of resources, it is not possible to claim one method's superiority over the other. We are rather highlighting our method's (1) annotation efficiency and (2) ability to extend support for newer languages (e.g., at present `GoogleLangID` has limited support for Odia (*or*) and no support for Assamese *as*).

7 Results and Analysis

Method	Overall accuracy	Excluding Odia
$\mathcal{F}_{\textit{end-to-end}}$	**99.4**	**99.8**
`GoogleLangID`	84.1	93.4

Table 4: Performance comparison. Since it is unclear if `GoogleLangID` supports Odia (*or*), the left-most column presents performance excluding Odia from the test set.

Table 4 summarizes the performance comparison between our proposed classifier and the `GoogleLangID` baseline. Our results indicate

that our method considerably outperforms the baseline which is a well-known commercial solution. A closer look at the individual performance of each language (confusion matrices presented in 2 and Table 3) reveals that across all languages, our method performs equal or better than `GoogleLangID`. Although the performance gap primarily stems from our method's overwhelmingly stronger performance in Odia (*or*), we perform considerably better than `GoogleLangID` even if we exclude Odia from the test data set.

Our performance comparison highlights the following two points. First, we reiterate that our goal is not to claim that our method would perform better than `GoogleLangID` across the board, but rather, to demonstrate the value our method adds in processing noisy social media texts. It is possible that our method is more attuned to noisy short social media texts while `GoogleLangID` could be (possibly) trained on cleaner corpora which explains our method's stronger performance. This is further corroborated by the fact that even our mispredictions (barring one) remain confined to the regional languages while many of `GoogleLangID`'s mispredictions are distributed across other languages (*ol*). Second, `GoogleLangID`'s weak performance in detecting Odia highlights the gap in current solutions and shows how our method can effectively and efficiently address these issues[2].

Recall that $\mathcal{D}_{tatoeba}$ contains a large set of languages (including the native script versions of the Indian languages considered in this paper). Test accuracy on a held-out set was 98.4%. Experiments reveal that introduction of the weakly labeled corpus does not impact performance on $\mathcal{D}_{tatoeba}$ (identical test accuracy of 98.4%).

7.1 Extensibility

We constructed a new data set of comments on YouTube videos from an Assamese news channel (News18 Assam/Northeast) and used the same approach of using $\hat{\mathcal{L}}_{polyglot}$ to obtain weak labels for Romanized Assamese. We do not show a direct comparison with `GoogleLangID` because `GoogleLangID` does not support Assamese (*as*). However, on an augmented test data set of 1,100 comments (100 Assamese comments with consensus labels from two annotators), we achieved a performance of 92% accuracy on identifying Assamese while retaining our previous performance on every other language. Assam has been a center for political debates and unrest in recent times (BBC). Our resource to detect Romanized Assamese can be a vital tool which to the best of our knowledge, does not exist. Details are presented in the Appendix.

[2]Odia is listed as one of the supported languages by `GoogleLangID`, it is unclear if this tool supports Romanized Odia. Assamese is not supported by `GoogleLangID`.

7.2 Domain-robustness

Our goal is to present an important Indian NLP resource that can perform well across multiple social media platforms. Hence, it is paramount that our system generalizes well both to *in domain* and *out of domain* instances. In the context of the task of language identification, domain adaption has received recent attention (Li et al., 2018). In this section, we present an analysis on our system's *out of domain* performance.

Data set of Hinglish tweets: We consider a data set of tweets introduced in Mathur et al. (2018). The data set consists of 3,189 tweets written in English (*en*), Romanized Hindi (*hi*) and code-mixed English-Hindi (*en-hi*). We construct a randomly sampled data set of 100 tweets with equal proportion of Hindi and English tweets (consensus labels obtained from two annotators).

As shown in Table 5, our system's *out of domain* performance was consistent with its *in domain* performance. We performed marginally better than `GoogleLangID`. We admit that a more robust test on multiple data sets comprising content from a larger set of Indian languages from other social media platforms would further validate our *out of domain* performance. However, our current experiment indicates that our system's success is not limited to YouTube comment texts, it can generalize to tweets as well.

7.3 Usage Statistics

As demonstrated, our integrated setup covers the Romanized and native script variants of the most prevalent Indian languages. This enables us to

Method	Accuracy	Language	P	R	F1
$\mathcal{F}_{end\text{-}to\text{-}end}$	**0.98**	*en*	**1.00**	**1.00**	**1.00**
		hi	0.96	**0.96**	**0.96**
`GoogleLangID`	0.95	*en*	**1.00**	**1.00**	**1.00**
		hi	**1.00**	0.90	0.95

Table 5: Performance comparison on the tweet data set. Best metric is highlighted in bold for each language. P: precision, R: recall.

State	*hi*	$hi_{\mathcal{N}}$	$hi \cap hi_{\mathcal{N}}$
Andhra Pradesh	1.66%	0.05%	1.71%
Bihar	67.97%	14.29%	82.26%
Gujarat	24.23%	3.41%	27.64%
Karnataka	1.85%	0.02%	1.87%
Kerala	0.48%	0.02%	0.5%
Madhya Pradesh	76.21%	10.39%	86.60%
Maharashtra	7.46%	4.18%	11.64%
Odisha	9.20%	0.02%	9.22%
Rajasthan	58.48%	29.90%	88.38%
Tamil Nadu	0.22%	0.01%	0.23%
Uttar Pradesh	63.56%	22.23%	85.79%
West Bengal	4.72%	0.18%	4.90%

Table 6: Presence of Hindi. *hi* is Romanized Hindi. $hi_{\mathcal{N}}$ is Devanagari Hindi. Hindi belt states (Jaffrelot, 2000) are highlighted with blue.

explore research questions on the usage patterns of these Indian languages. This part of our analysis is conducted on the entire election corpus containing comments written in all scripts.

Weak geo-labels: Recall that, all of the 24 regional news outlets we consider present news in the dominant language of their respective states. Hence, it is reasonable to assume that a considerable fraction of users consuming the regional news and participating in the comments section have some affiliation to the region (state). Thus, a comment posted in response to a regional news outlet's video can be assigned a weak/noisy geographic label - the state targeted by the news network. For instance, we assume that a comment posted in response to a Tamil news video clip, is likely to be authored by someone who either resides in or retains strong ties to Tamil Nadu. Combining these weak/noisy geographic labels with our language identification system, we can assess the geographic distribution of language use in India. Note that, these results are only approximate estimates - YouTube comments do not contain any geographic information. Further, it is also not possible to estimate a user's knowledge of other languages through our analysis. For example, if a user comments solely in Hindi, it is not possible to assess their fluency in English or Bengali.

7.4 Hindi Web Usage

Geographic extent of Hindi usage: We label each comment with the language prediction from $\mathcal{F}_{end\text{-}to\text{-}end}$ and a geographic label corresponding to the origin state of the news outlet. All comments posted in Romanized and Devanagari Hindi (denoted by *hi* and $hi_{\mathcal{N}}$, respectively) are retained and the resulting choropleth is visualized in Figure 1(b). We also provide in Figure 1(a), the region referred to as the Hindi belt (Jaffrelot, 2000) where Hindi is the first language of the bulk of the population. We observe a strong correlation of the estimated geographic extent of Hindi with the Hindi belt states.

In Table 6, we list the state-wise estimates of Hindi usage in our corpus. Our findings are consistent with existing knowledge of Hindi's geographic spread. In the Hindi belt states, more than 80% of the comments were authored in Hindi. Consistent with census data (Census, 2011) and prior literature (Ramaswamy, 1997), the fraction of Hindi comments discovered in the Tamil Nadu origin subset was minuscule.

Romanized vs Devanagari: As shown in Table 6, the ratio of *hi* and $hi_{\mathcal{N}}$ usage reveals that a vast majority of internet users eschew the traditional Devanagari script and instead use Roman script. However, the ratio of Roman script to Devanagari script is substantially less lopsided in the Hindi belt states than in the other states. Our studies are consistent with Gella et al. (2014).

Estimating bilinguality: We conduct a user-focused study by computing language usage statistics on a per-user basis. We assume that a user is proficient in a language, $\mathcal{L}$, if she posts two or more comments (in order to accommodate for some estimation error) in $\mathcal{L}$. If a user is estimated to be proficient in two languages, then we label her as bilingual. Romanized and native script comments are both considered to be an equal demonstration of proficiency in a given language. We acknowledge that this is at best a noisy estimate.

Out of 159,993 total users, 41,776 users (26.1%) were marked as bilinguals using $\mathcal{F}_{end\text{-}to\text{-}end}$. According to the 2011 census (Census, 2011), 26% of the Indian population are bilinguals. Hence, surprisingly, our noisy estimate was reasonably close to the ground truth. We observe that over 70% of the discovered bilinguals in our corpus used Hindi-English. A detailed plot is presented in the Appendix.

7.5 Trace Language Detection

We conclude this section with an analysis on (1) to what extent we address $\hat{\mathcal{L}}_{polyglot}$'s inability to detect trace languages, and (2) why it could be worth addressing. Our definition of trace language is corpus-specific. We consider a language $\mathcal{L}$ to be a trace language in a corpus $\mathcal{D}$ if fewer than 1%

(a) Hindi Belt States (b) Total Hindi Usage per State (c) Legend

Figure 1: Choropleths of Hindi usage patterns in India. (a) shows the geographic region identified as the Hindi belt. (b) shows the patterns of Hindi usage. We extend the results for Andhra Pradesh to Telangana, and Bihar to Jharkhand because the same news networks cater to both states. The base maps used for this plot are sourced from the Government of India. The authors are aware that these maps include disputed territories. These maps do not constitute judgments on existing disputes.

Language	Data set	Example comment	Loose translation
bn	$\mathcal{D}_{hope}$	**Indian air force k varote ferea deoar jonno osonkho dhonnobad Pak armyke**	*Thank you Pak army for returning our Indian Air Force (pilot).*
te	$\mathcal{D}_{hope}$	**yudham vasthey manam kuda chala nastha potham...**	*We'll suffer a lot too in the event of war...*
bn	$\mathcal{D}_{help}$	**Tawhid ami rohiggha dhar help korte chai plz amaka hepl korar moto kisu opai bolo bz ami dhashar bahira thaki**	*Tawhid, I want to help the Rohingyas. Please suggest me some ways I can help them. I live outside of the country.*
te	$\mathcal{D}_{help}$	**manama hinduvulama muslim christans ani kadu manushulama aannadi kavali manamu valla paristutulalo unte telustundi**	*Are we Hindu, Muslim, Christian - this is irrelevant, are we human - this is relevant; if we were in their position, we would understand*
bn	$\mathcal{D}_{COVID}$	**Amra 500jon Lok Bangalore atkie Achi please amader Bari. Pouche din ,amader Bari west Bengal India**	*We are 500 people stuck in Bangalore. Please make some arrangements so that we can reach our home in West Bengal, India.*
te	$\mathcal{D}_{COVID}$	**Please sir twitter lo pettandi please nenoka mahilalu 2 chinna pillalu unnaru Sir mem jammikunta lo undipoyamu nijamabad cherela chudandi Sir**	*Please sir, post on Twitter, I'm a woman with 2 small kids, we're stuck in Jammikunta, please help us get to Nijambad*

Table 7: Random sample of comments in trace language detected by our system.

of the documents in $\mathcal{D}$ are authored in $\mathcal{L}$. In this section, we focus on the following three corpora one of which ($\mathcal{D}_{COVID}$) we introduce here:

1. $\mathcal{D}_{hope}$: 2.04 million YouTube comments relevant to the 2019 India-Pakistan conflict (Palakodety et al., 2020a).

2. $\mathcal{D}_{help}$: 263k YouTube comments relevant to the Rohingya refugee crisis (Palakodety et al., 2020c).

3. $\mathcal{D}_{COVID}$: 777,748 comments from 5,301 videos from two highly popular Indian news channels (NDTV and Zee News) posted between 30th January, 2020[3] and 10th April, 2020.

As reported in Palakodety et al. (2020a), $\hat{\mathcal{L}}_{polyglot}$ discovered three clusters in $\mathcal{D}_{hope}$: (1) *en*, (2) *hi* and (3) $hi_{\mathcal{N}}$; no other languages were detected. However, in our experiments, we found presence of multiple trace languages. For instance, overall, our method $\mathcal{F}_{end\text{-}to\text{-}end}$ found 3,373 Telugu (*te*) and 205 Bengali (*bn*) comments in $\mathcal{D}_{hope}$. Human annotation of randomly sampled 100 comments in each of the two languages revealed a precision of 100% and 97% for *te* and *bn*, respectively. Similarly, we conducted a search for *bn* and *te* comments in $\mathcal{D}_{help}$, and found 1,251 and 146 comments, respectively. Human annotation on randomly sampled 100 comments from each of the two languages yielded

[3]First COVID-19 positive case was reported in India on this day.

precision of 99% for both *bn* and *te*. In Table 7, we list example peace-seeking, hostility-diffusing comments (*hope speech*) and comments indicating support for the disenfranchised Rohingyas (*help speech*).

Finally, when $\mathcal{F}_{end\text{-}to\text{-}end}$ is run on $\mathcal{D}_{COVID}$, we discover comments in several languages requesting assistance during the nationwide lockdown (BBC, 2020). Our method reveals the presence of vulnerable individuals who express themselves in low-resource languages. We hope our tool can open the gates for research in this humanitarian domain.

8 Conclusion

In this paper, we present a language identification tool with a focus on nine major Romanized Indian languages. Despite the widespread use of Romanization on social media, NLP resources and tools often focus more on the native scripts. Our tool integrates with an existing large-scale corpus and holds promise in being a valuable resource for Indian social media analysis. Our pipeline leverages a recent NLP algorithm and obtains weak labels for a large number of samples substantially reducing the annotation cost. Finally, we conduct studies on the geographic extent, bilinguality, and Romanization of Hindi and observe that these align with existing studies and surveys.

References

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*, pages 13–23.

BBC. Why has india's assam erupted over an 'anti-muslim' law? Online; accessed 12-May-2020.

BBC. 2020. Coronavirus: India's pandemic lockdown turns into a human tragedy. Online; accessed 3-June-2020.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific twitter collections. In *Proceedings of the second workshop on language in social media*, pages 65–74. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Census. 2011. 2011 census data. Online; accessed 3-June-2020.

Arunavha Chanda, Dipankar Das, and Chandan Mazumdar. 2016. Unraveling the English-Bengali code-mixing phenomenon. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 80–89, Austin, Texas. Association for Computational Linguistics.

Monojit Choudhury, Kalika Bali, Tirthankar Dasgupta, and Anupam Basu. 2010. Resource creation for training and testing of transliteration systems for indian languages. LREC.

Matteo Cinelli, Walter Quattrociocchi, Alessandro Galeazzi, Carlo Michele Valensise, Emanuele Brugnoli, Ana Lucia Schmidt, Paola Zola, Fabiana Zollo, and Antonio Scala. 2020. The covid-19 social media infodemic.

Ma Dewen. 2010. A study on the two waves of states-reorganization in india [j]. *South Asian Studies Quarterly*, 1.

Pinar Donmez, Jaime Carbonell, and Jeff Schneider. 2010. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 826–837. SIAM.

FastText. FastTextLangID. [Online; accessed 3-June-2020].

Spandana Gella, Kalika Bali, and Monojit Choudhury. 2014. "ye word kis lang ka hai bhai?" testing the limits of word level language identification. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 368–377.

Google. GoogleLangID. [Online; accessed 3-June-2020].

HindustanTimes. 2019. Youtube now has 265 million users in india. Online; accessed 3-June-2020.

Aaron Jaech, George Mulcaire, Shobhit Hathi, Mari Ostendorf, and Noah A. Smith. 2016. Hierarchical character-word models for language identification. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, pages 84–93, Austin, TX, USA. Association for Computational Linguistics.

Christophe Jaffrelot. 2000. The rise of the other backward classes in the hindi belt. *The Journal of Asian Studies*, 59(1):86–108.

Tommi Sakari Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Pratik Joshi, Christain Barnes, Sebastin Santy, Simran Khanuja, Sanket Shah, Anirudh Srinivasan, Satwik Bhattamishra, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali. 2019. Unsung challenges of building and deploying language technologies for

low resource language communities. *arXiv preprint arXiv:1912.03457*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th EACL: Volume 2, Short Papers*, pages 427–431.

Ashiqur R. KhudaBukhsh, Shriphani Palakodety, and Jaime G. Carbonell. 2020. Harnessing code switching to transcend the linguistic barrier. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4366–4374. ijcai.org.

Upendra Kumar, Vishal Singh, Chris Andrew, Santhoshini Reddy, and Amitava Das. 2018. Consonant-vowel sequences as subword units for code-mixed languages. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014. Sata-anuvadak: Tackling multiway translation of indian languages. *pan*, 841(54,570):4–135.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. What's in a domain? learning domain-robust text representations using adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 474–479, New Orleans, Louisiana. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of twitter messages. In *Proceedings of the 5th workshop on language analysis for social media (LASM)*, pages 17–25.

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148, Brussels, Belgium. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations*.

Tom M Mitchell. 2004. The role of unlabeled data in supervised learning. In *Language, Knowledge, and Representation*, pages 103–111. Springer.

Gaurav Mohanty, Abishek Kannan, and Radhika Mamidi. 2017. Building a sentiwordnet for odia. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 143–148.

Shriphani Palakodety, Ashiqur R. KhudaBukhsh, and Jaime G. Carbonell. 2020a. Hope speech detection: A computational analysis of the voice of peace. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 1881–1889. IOS Press.

Shriphani Palakodety, Ashiqur R. KhudaBukhsh, and Jaime G. Carbonell. 2020b. Mining insights from large-scale corpora using fine-tuned language models. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 1890–1897. IOS Press.

Shriphani Palakodety, Ashiqur R. KhudaBukhsh, and Jaime G. Carbonell. 2020c. Voice for the voiceless: Active sampling to detect comments supporting the rohingyas. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 454–462.

Sumathi Ramaswamy. 1997. *Passions of the tongue: Language devotion in Tamil India, 1891–1970*, volume 29. Univ of California Press.

Rupak Sarkar, Sayantan Mahinder, Hirak Sarkar, and Ashiqur R. KhudaBukhsh. 2020. Social media attributions in the context of water crisis. In *Empirical Methods in Natural Language Processing (EMNLP), 2020*, page to appear.

Statista. 2020. Most popular social networks worldwide as of january 2020, ranked by number of active users. Online; accessed 3-June-2020.

Tatoeba. 2020. Tatoeba. Online; accessed 3-June-2020.

R Vijayakrishna and L Sobha. 2008. Domain focused named entity recognizer for tamil using conditional random fields. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition-Volume 15*, pages 57–64. Association for Computational Linguistics.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. POS tagging of English-Hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979, Doha, Qatar. Association for Computational Linguistics.

YourStory. 2018. Youtube monthly user base touches 265 million in india, reaches 80 pc of internet population. Online; accessed 3-June-2020.

9 Appendix

9.1 Annotation

All annotations are performed by two native speakers of each of the languages we considered. All labels are consensus labels.

9.2 Detailed Performance with Assamese

Our data set was crawled using publicly available YouTube API on the YouTube channel of CNN News18 Assam/Northeast. Overall, we obtained 66,923 comments from 7,170 videos of which weak labels (4,337 English and 21,411 Assamese) were obtained using $\hat{\mathcal{L}}_{polyglot}$. We augmented our previous training set with these obtained (weak labels) comments. The detailed performance is presented in Table 9.

9.3 Classification Framework

State	Channels
Andhra Pradesh	V6 News Telugu TV9 Telugu Live
Bihar	News18 Bihar Jharkhand ZeeBiharJharkhand
Gujarat	TV9 Gujarati ABP Asmita
Karnataka	TV9 Kannada Suvarna News
Kerala	Asianetnews Manorama News
Madhya Pradesh	News18 MP Chhattisgarh Zee Madhya Pradesh Chhattisgarh
Maharashtra	ABP Majha ZEE 24 TAAS
Odisha	OTV News18 Odia
Rajasthan	News18 Rajasthan ZeeRajasthanNews
Tamil Nadu	Puthiyathalaimurai TV Polimer News
Uttar Pradesh	News18 UP Uttarakhand Zee Uttarpradesh Uttarakhand
West Bengal	ABP ANANDA News18 Bangla

Table 8: Regional channels.

The classification framework we use (Joulin et al., 2017), contains a variety of optimizations focused on text classification - an architecture that enables parameter sharing, and efficient techniques to include token n-grams. The inference phase is able to process and label over 10 million documents in under five minutes (wall clock time).

9.4 Test data set details

100 comments are randomly sampled for each of the 10 languages (*bn*, *en*, *gu*, *hi*, *kn*, *ml*, *mr*, *or*, *ta*, *te*). The average number of tokens in the comments is 22.6 $\pm$ 18.7. A language-wise breakdown is presented in Table 10.

		Predicted Label										
		as	*bn*	*en*	*gu*	*hi*	*kn*	*ml*	*mr*	*or*	*ta*	*te*	*ol*
True Label	*as*	92	3	0	1	2	0	0	0	2	0	0	0
	bn	0	100	0	0	0	0	0	0	0	0	0	0
	en	0	0	100	0	0	0	0	0	0	0	0	0
	gu	0	0	0	100	0	0	0	0	0	0	0	0
	hi	0	0	0	0	100	0	0	0	0	0	0	0
	kn	1	0	0	0	0	99	0	0	0	0	0	0
	ml	0	0	0	0	0	0	99	0	0	1	0	0
	mr	0	0	0	0	0	0	0	100	0	0	0	0
	or	0	0	0	1	2	0	0	0	96	1	0	0
	ta	0	0	0	0	0	0	0	0	0	100	0	0
	te	0	0	0	0	0	0	0	0	0	0	100	0
	ol	0	0	0	0	0	0	0	0	0	0	0	0

Table 9: Confusion matrix of performance evaluation of $\mathcal{F}_{end\text{-}to\text{-}end}$ on 1,100 annotated comments; *ol* denotes **o**ther **l**anguages.

9.5 Language pairs used by bilinguals

Figure 2 summarizes the relative distribution of language pairs in our bilingualism estimation experiment. Results show that Hindi-English bilingualism is the most dominant one.

State	Comment length
as	17.03 $\pm$ 19.95
bn	25.19 $\pm$ 18.33
en	31.52 $\pm$ 27.04
gu	23.82 $\pm$ 20.91
hi	24.72 $\pm$ 15.33
kn	17.85 $\pm$ 10.57
ml	19.91 $\pm$ 11.92
mr	25.73 $\pm$ 21.88
or	13.34 $\pm$ 11.72
ta	21.98 $\pm$ 14.59
te	22.05 $\pm$ 20.95

Table 10: Statistics of test data set.

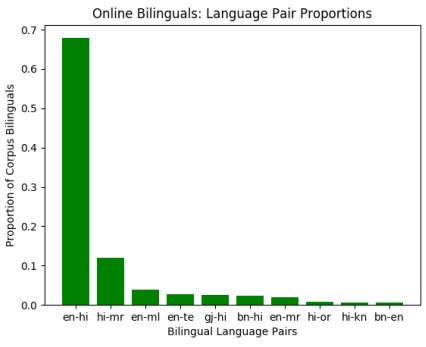

Figure 2: The top language pairs used by bilinguals in our corpus. Hindi and English feature prominently in all the language pairs.

IndiaTV, NDTV India, Republic World, The Times of India, Zee News, Aaj Tak, ABP NEWS, CNN-News18, News18 India, NDTV, TIMES NOW, India Today, The Economic Times, Hindustan Times

Table 11: National channels.

9.6 List of YouTube channels

The YouTube channels considered in Palakodety et al. (2020b) are listed in Table 8 and 11.