hinglishNorm - A Corpus of Hindi-English Code Mixed Sentences for Text Normalization

Piyush Makhija
piyush@vahan.co

Ankit Kumar
ankit@vahan.co

Anuj Gupta
anuj@vahan.co

Abstract

We present *hinglishNorm* - a human annotated corpus of Hindi-English code-mixed sentences for text normalization task. Each sentence in the corpus is aligned to its corresponding human annotated normalized form. To the best of our knowledge, there is no corpus of Hindi-English code-mixed sentences for text normalization task that is publicly available. Our work is the first attempt in this direction. The corpus contains 13494 segments annotated for text normalization. Further, we present baseline normalization results on this corpus. We obtain a Word Error Rate (WER) of 15.55, BiLingual Evaluation Understudy (BLEU) score of 71.2, and Metric for Evaluation of Translation with Explicit ORdering (METEOR) score of 0.50.

1 Introduction

Hindi is the fourth most-spoken first language in the world[1]. According to one estimate, nearly 0.615 billion people speak Hindi as their first language[2]. Of these people, most of the speakers are in India. The second most spoken language in India is English[3]. Hindi and English are the official languages of the Indian Commonwealth[4]. A large number of these people have joined the Internet recently. As a matter of fact, Next Billion Users (NBU) is a term commonly used in tech and business circles to refer to the large number of people from India, Brazil, China and South-East Asia who joined the Internet in the last decade[5]. This phenomena is primarily attributed to ubiquitous highly affordable phone and internet plans[6]. A large fraction of NBU users come from India and speak Hindi as either their first or second language. A large number of these people use a blend of Hindi and English in their daily informal communication. This hybrid language is also known as Hinglish[7].

These users extensively use Internet platforms which heavily rely on User Generated Content (UGC) - social media platforms such as Facebook or Twitter; messaging platforms such as WhatsApp or Facebook messenger; user reviews aggregators such as the Google play store or Amazon. A key characteristic of their behaviour on such platforms is their use of Hinglish. Thus, building any Natural Language Processing (NLP) based Internet applications for these users necessitates the ability to process this 'new' language. Further, these UGC platforms are notoriously noisy. This means there is an additional challenge of non-canonical text. Therefore, a key step in building applications for such text data is *text normalization*. Intuitively, it is transforming text to a form where written text aligned to its normalized spoken form (Sproat and Jaitly, 2016). More formally, *it is the task of mapping non-canonical language,*

[1] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

[2] https://blog.busuu.com/most-spoken-languagesintheworld/

[3] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India

[4] https://en.wikipedia.org/wiki/Hindi

[5] https://www.blog.google/technology/nextbillionusers/nextbillionusersarefutureinternet/

[6] https://www.hup.harvard.edu/catalog.php?isbn=9780674983786

[7] https://en.wikipedia.org/wiki/Hinglish

Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, pages 136–145
Barcelona, Spain (Online), December 12, 2020

typical of speech transcription and computer-mediated communication, to standardized writing (Lusetti et al., 2018).

Separately, there has been a lot of work in the two areas of normalization and building corpora of Hindi-English code mix text data, not much has been done at the intersection of the two(refer to section 2). To the best of our knowledge, there does not exist a corpus of Hindi-English Code Mixed sentences for normalization where the normalizations are human annotated. This work is an effort to release such a corpus

This work is motivated from our business use case where we are building a conversational system over WhatsApp to screen candidates for blue-collar jobs. Our candidate user base often comes from tier-2 and tier-3 cities of India. Their responses to our conversational bot are mostly a code mix of Hindi and English coupled with non-canonical text (ex: typos, non-standard syntactic constructions, spelling variations, phonetic substitutions, foreign language words in non-native script, grammatically incorrect text, colloquialisms, abbreviations, etc). The raw text our system gets is far from clean well formatted text and text normalization becomes a necessity to process it any further.

The main contributions of this work are two-fold, viz. (i) creating a human annotated corpus for text normalization of Hindi-English code mix sentences; and (ii) reporting baseline metrics on the corpus. Further, we release the corpus and annotations under a Creative Commons Attribution-NonCommercial-ShareAlike License[8].

2 Related Work

In this section, we present relevant work in the following areas viz.(1) Text Normalization (2) Normalization and UGC Datasets (3) Code-mixed Datasets, (4) Hindi-English Datasets.

Text Normalization: Text normalization, sometimes also called lexical normalization, is the task of translating/transforming a non-standard text to a standard format. Using text normalization on noisy data, one can provide cleaner text data to downstream NLP tasks and improve the overall system performance (Liu et al., 2012), (Satapathy et al., 2017). Some of the early work used a rule-based spell-checker approach to generate a list of corrections for any misspelled word, ranked by corresponding posterior probabilities (Church and Gale, 1991) (Mays et al., 1991) (Brill and Moore, 2000). However, this approach did not factor in any context while normalizing words. (Choudhury et al., 2007) used a Hidden Markov Model (HMM), where they modeled each standard English word as a HMM and calculated the probability of observing the noisy token. "Moses", a well known Statistical Machine Translation (SMT) tool, provided significant improvements in comparison to previous solutions (Koehn et al., 2007). (Aw et al., 2006) adapted a phrase-based Machine Translation (MT) model for normalizing SMS and achieved significant gain in performance. In the past few years, Neural network based approaches for text normalization have become increasingly popular and have shown competitive performance in shared tasks (Chrupała, 2014), (Min and Mott, 2015). (Lusetti et al., 2018), (Liu et al., 2012) and (Satapathy et al., 2017) provide excellent literature covering the landscape on this topic.

Normalization and UGC Datasets: (Han and Baldwin, 2011) introduced a text normalization approach for twitter data using a variety of supervised & unsupervised learning techniques. This study resulted in 'lexNorm'[9], an open-source dataset containing 549 tweets. (Baldwin et al., 2015) subsequently released lexNorm15[10]. This new dataset contained 2950/1967 annotated tweets in train/test sets. (Michel and Neubig, 2018) created the MTNT dataset[11] containing translations of Reddit comments from the English language to French/ Japanese and vice versa, containing 7k∼37K data points per language pair. This dataset contains user-generated text with different kinds of noise, e.g., typos, grammatical errors, emojis, spoken languages, etc. for two language pairs. (van der Goot and van Noord, 2017) introduced 'MoNoise', a general purpose model for normalizing UGC text data. This model utilizes Aspell spell checker, an n-gram based language model and word embeddings trained on a few million tweets. It

[8] http://creativecommons.org/licenses/by-nc-sa/4.0/

[9] http://people.eng.unimelb.edu.au/tbaldwin/etc/lexnorm_v1.2.tgz

[10] https://github.com/noisy-text/noisy-text.github.io/blob/master/2015/files/lexnorm2015.tgz

[11] https://www.cs.cmu.edu/ pmichel1/mtnt/

Dataset	Task	Size
IITB English-Hindi Parallel Corpus (Anoop et al., 2018)	Machine Translation	Train - 1,561,840 Dev - 520 Test - 2,507
HindiEnCorp 0.5 (Dhariya et al., 2017)	Machine Translation	132,300 sentences
Xlit-Crowd: Hindi-English Transliteration Corpus (Khapra et al., 2014)	Machine Translation	14,919 words
IIITH Codemixed Sentiment Dataset (Prabhu et al., 2016)	Sentiment Analysis	4,981 sentences

Table 1: indicnlp_catalog Hindi-English Datasets

gave significant improvement in State-Of-The-Art (SOTA) normalization performance on the lexNorm15 dataset. (Muller et al., 2019) focused on enhancing BERT model on UGC by applying lexical normalization.

Code-Mixed Datasets: Since the launch of EMNLP Shared Tasks of Language identification in Code-Switched Data[12], there has been an increased focus on analyzing the nature of code-mixed data, language identification approaches and how to carry out NLP tasks like POS tagging and Text normalization on such text data. For the first shared task, code-switched data was collected for language pairs such as Spanish-English (ES-EN), Mandarin-English (MAN-EN), Nepali-English(NEP-EN) and Modern Standard Arabic - Dialectal Arabic(MSA-DA) (Solorio et al., 2014). Subsequently more language pairs were added with primary focus on language identification task (Molina et al., 2019). (Aguilar et al., 2019) introduced Named Entity Recognition on Code-Switched Data. (Mandal et al., 2018) introduced Bengali-English code-mixed corpus for sentiment analysis. More recently, normalization of code mixed data has been receiving a lot of attention. (Barik et al., 2019) worked on normalizing Indonesian-English code-mixed noisy social media data. Further, they released 825 annotated tweets from this corpus[13]. (Phadte and Thakkar, 2017) focused on normalization of Konkani-English code-mixed text data from social media. (Adouane et al., 2019) worked on normalizing algerian code-switched UGC utilizing encoder-decoder network and showed promising results.

Hindi-English Datasets: (Vyas et al., 2014) was one of the earliest work to focus on creating a Hindi-English code-mixed corpus from social media content for POS tagging. The same year (Bali et al., 2014) analyzed Facebook English-Hindi posts to show a significant amount of code-mixing. (Bhat et al., 2018) worked with similar English-Hindi code-mixed tweets in roman script for dependency parsing. (Patra et al., 2018) worked on sentiment analysis of code mixed Hindi-English & Bengali-English language pairs. (Singh et al., 2018) focused on normalization of code-mixed text using pipeline processing to improve the performance on POS Tagging task. indicnlp_catalog[14] is a effort to consolidate resources on Indian languages. Table 1 presents the most relevant Hindi-English datasets from this effort.

While there exists extensive work in each of these areas, for some reason normalization of Hindi-English (which is at intersection of these areas) hasn't received its due attention. This may be partly due to unavailability of a comprehensive data set and baseline. We believe our work will address this lacuna.

3 Corpus Preparation

While preparing this corpus, we carry out the following steps.

1. **Data Collection**: collecting Hindi-English sentences.
2. **Data Filtering & Cleaning**: standard pre-processing of raw sentences.

[12]http://emnlp2014.org/workshops/CodeSwitch/call.html
[13]https://github.com/seelenbrecher/code-mixed-normalization/tree/master/data
[14]https://github.com/anoopkunchukuttan/indic_nlp_library

3. **Data Annotation**: sentence-level text normalization by human annotators.

3.1 Data Collection

We collected data in two phases: In the first phase we built and deployed general chit-chat bots on social media platforms. User responses were randomly sampled and pooled to create the dataset. In the second phase, we collected data from our platform. Here too the responses were chosen randomly to be added to the dataset.

3.2 Data Filtering & Cleaning

The raw text data we collected was then preprocessed and cleaned. Following were the key steps:

1. Drop all messages that were forwarded messages or consisted of only emojis.
2. Hindi words were written in both scripts - Devanagari and Roman. All words in Devanagari were converted into roman script.
3. Removed all characters other than alpha-numeric or white space.
4. All sentences containing profane words or phrases were dropped.
5. All sentences containing any Personal Identification Information (PII) were dropped.

The exact steps followed can be found here[15]. Steps (4) and (5) were done manually.

3.3 Data Annotation

The preprocessed data was sent to human annotators for text normalization annotation. Each word in the input sentence was tagged for the type of non-canonical variation & its phonetically standard transliteration[16]. The annotators chosen were native speakers of Hindi and had bilingual proficiency in English. The dataset was annotated by three annotators while maintaining an average error rate of less than 5% on the dataset.

Based on the context in which the word appears in the input sentence, annotators provide the corresponding normalized word. Further, to better capture the process used by the annotators to arrive at the normalized text, the annotators provide a unique *tag* for each word. This tag describes the transformation applied by annotators to arrive at the corresponding normalized word. The corpus along with normalized text also contains these tags. (van der Goot et al., 2018) proposes a taxonomy to annotate normalization of UGC in parallel sentences. We follow a similar but independent approach. Below we describe various tags used in the corpus, the scenario in which a given tag is used and explain the transformation applied with example(s):

1. **Looks Good**: The word under consideration is already an English word with proper spelling. No corrective action is required here. e.g. *"yes"*, *"hello"*, *"friend"*.
2. **Merge**: A word is mistakenly split into two or more consecutive words by uncautious white spaces. In such cases, the corrective action is to merge such words. e.g. *"ye s"* → *"yes"*, *"hell oo"* → *"hello"*, *"fri en dd"* → *"friend"*.
3. **Split**: Two words get conjoined or when a user uses a contraction of two words. In such cases, the corrective action is to split the words with correct spelling. e.g. *"yeshellofriend"* → *"yes hello friend"*, *"isn't"* → *"is not"*, *"should've"* → *"should have"*
4. **Short Form**: The word is a short form (phonetically or colloquially). In such cases the corrective action is to replace the word with the corresponding full form. e.g. *"u"* → *"you"*, *"y"* → *"why"*, *"doc"* → *"doctor"*.
5. **Acronym**: The word is an acronym or abbreviation. In such cases, the corrective action is to replace with their full form. e.g. *"fb"* → *"facebook"*, *"brb"* → *"be right back"*

[15] https://github.com/piyushmakhija5/normalizationDataset/blob/master/dataPreprocessing.py
[16] https://www.iso.org/standard/28333.html

6. **Typo**: The word is a typo/spelling mistake if its spelling is incorrect. This is an unintentional error (due to haste, fat-finger error[17] or low attention to details) made while typing. In such cases the corrective action is to undo the typing error. e.g *"yess"* → *"yes"*, *"helllo00o"* → *"hello"*, *"frieendd"* → *"friend"*
7. **Wordplay**: User has deliberately modified the word for creative purposes. In such cases, the corrective action is to undo the creativity and replace with correct spellings. e.g. *"hiiiii"* → *"hi"*, *"I 8t mahh food"* → *"I ate my food"*
8. **Profanity**: The word is a profane word. In such cases, no corrective action is taken. e,g, *"f**k"*, *"ba*tard"*.
9. **Hindi**: The word is a Hindi language word in Roman script. In case the spelling is incorrect, replace the word with the corresponding phonetically correct transliteration. e.g. *"haaan"* → *"haan"*[18], *"namskar"* → *"namaskaar"*[19]
10. **Unrecognizable or Other Language (UOL)**: Default tag to catch words from any language other than English and Hindi or any other type of error/variation that is not covered by any of the previously mentioned tags such as gibberish. In such cases, no corrective action is taken. e.g.*"asgjhsad is my name"*, *"n233s is cool"*, *"call maadi"*[20]

3.4 Corpus Examples

To facilitate better understanding, we list out two examples from our dataset to illustrate the input text and corresponding normalized text along with annotations:

1. **id:** 5089
   inputText: *"pakka lisenc nahin hai but ll hai"*
   tags: ['Hindi', 'Typo', 'Hindi', 'Hindi', 'Looks Good', 'Acronym', 'Hindi']
   normalizedText: *"pakka license nahi hai but learning licence hai"*[21]

   Here, the word 'pakka' is a Hindi word that means 'confirmed'. 'lisenc' is a typo for 'license' and refers to driving license. 'nahin' in hindi means 'no', so on and so forth.

2. **id**: 13427
   inputText: *"hiii mjhe jab chaiye"*
   tags: ['Wordplay', 'Hindi', 'Typo', 'Hindi']
   normalizedText: *"hi mujhe job chaahie"*[22]

   Here, the word 'hiii' is a wordplay for 'hi', 'mjhe' is a typo for hindi word 'mujhe' which means 'I'. 'jab' is a typo for 'job' and 'chaiye' is typo for hindi word 'chaahie' which means 'want'.

4 Corpus Analysis

After the preprocessing and manual annotation as described in Section 3, we refer to the data set obtained as *hinglishNorm*. It contains 13494 sentences along with their text normalized form. Table 2 presents some basic statistics of the *hinglishNorm* corpus. Each data point in the corpus is a sentence pair consisting of an *inputText* and *normalizedText*. *inputText* is the text as given by the user after preprocessing and *normalizedText* is the corresponding human annotated text. Table 3 gives corpus level statistics of

[17] https://en.wikipedia.org/wiki/Fat-finger_error
[18] Hindi word corresponding to *"yes"* in English
[19] Hindi greeting corresponding to *"hi"* in English
[20] Slang that means *"call me"*
[21] Corresponding English translation: "don't have a permanent license, but I have learning licence"
[22] Corresponding English translation: *"hi, I want a job"*

Attribute	Value
# Datapoints	13494
# Train	10795
# Test	2699
% Sentences Modified after Annotation	80.08%
% Hindi-English Code-Mixing Sentences	52.69%
% Non-English/Non-Hindi words	5.41%
% Normalized Words in Corpus	54.25%
% Hindi Words in Corpus	41.48%
Code-Mixing Index (CMI) (Das and Gambäck, 2014)	88.40

Table 2: Basic Statistics *hinglishNorm* Corpus

Features	*inputText*	*normalizedText*
# Sentence	13494	13494
# Unique Sentences	13066	12547
# Unique Words	9326	7465
# Unique Characters	37	37
Most Common Sentence	"whats ur name"	"what is your name"
# Most Common Sentence	12	38
Mean Character Length	22.06	25.25
Std Var of Character Length	16.97	19.00
Median Character Length	18	21
Mean Word Length	4.96	5.13
Std Var of Word Length	3.53	3.66
Median Word Length	4	4

Table 3: Statistics for *inputText* vs *normalizedText*

inputText and *normalizedText*. Each of these data points is also annotated with *tags* which denotes the transformation applied to obtain the text normalized form of *inputText*. Figure 1 gives us distribution of tags within the dataset.

An important aspect of this corpus is that the correct normalized equivalent of an input word can vary. Based on the context in which the the input word appears in the sentence, the misspelled words might require different corrections. For e.g.

- *"hii, I have a bike" (inputText)* → *"hi, I have a bike" (normalizedText)*
  - Input text provided by the user is an English language sentence with misspelled *"hi"*. Annotators understand that the word belongs to English language and correct spelling, in this case, should be *"hi"*
- *"mere pass bike hii" (inputText)* → *"mere pass bike hai" (normalizedText)*
  - Input text is a romanized version of a Hindi sentence that means *"I have a bike"*. Annotators understand that the word belongs to Hindi language and correct spelling, in this case, should be *"hai"*

5 Benchmark Baseline

It is common to model the text normalization problem as a Machine Translation problem (Mansfield et al., 2019) (Lusetti et al., 2018) (Filip et al., 2006) (Zhang et al., 2019). Given that Bidirectional LSTM with attention is a popular baseline model for machine translation task, we built a text normalization model using the same on the lines of work by (Bahdanau et al., 2014). We evaluated our system using

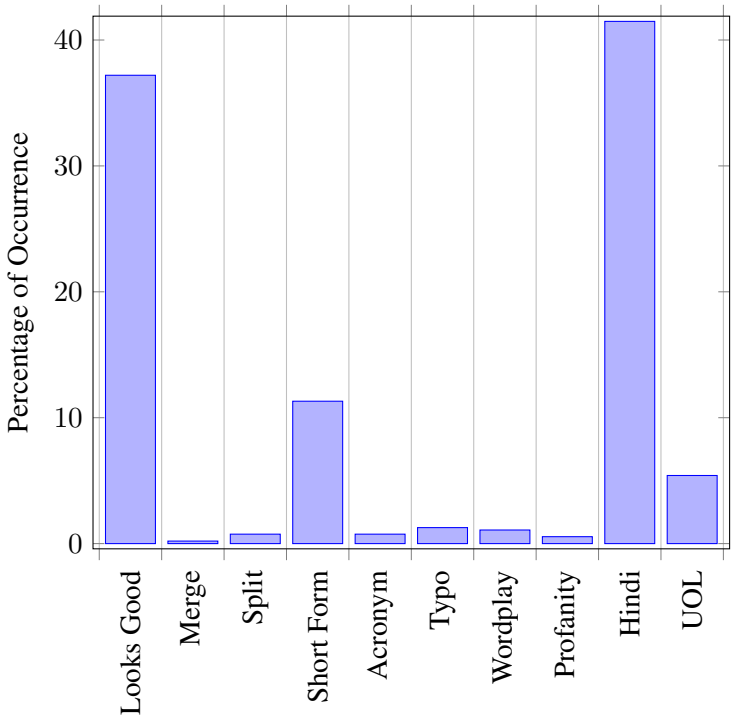

Figure 1: Distribution of tags in hinglishNorm Dataset

Evaluation Metric	Baseline
WER	15.55
BLEU	71.21
METEOR	0.50

Table 4: Baseline Performance on *hinglishNorm*

well established metrics - Word-Error Rate (WER) (Nießen et al., 2000), BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002) and Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Banerjee and Lavie, 2005). Table 4 shows the results of our experiments over *hinglishNorm*.

6 Availability

The homepage for the dataset can be accessed here[23].

The new corpora we release are available for research and non-commercial use under a Creative Commons Attribution-NonCommercial-ShareAlike License[24].

7 Conclusion & Future Work

We presented *hinglishNorm* version 1.0, a corpus of Hindi-English code mix sentences for text normalization task. Thereby, filling a much needed gap. The purpose of this corpus is to serve as a benchmark dataset for evaluation of Hindi-English code mixed text normalization model performance. We have also provided our benchmark baseline results on this corpus for comparison. As future work, we plan to build stronger baselines using SOTA models such as BERT (Devlin et al., 2018), DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), etc.

[23]https://github.com/piyushmakhija5/hinglishNorm
[24]http://creativecommons.org/licenses/by-nc-sa/4.0/

References

Wafia Adouane, Jean-Philippe Bernardy, and Simon Dobnik. 2019. Normalising non-standardised orthography in Algerian code-switched user-generated data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 131–140, Hong Kong, China, November. Association for Computational Linguistics.

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2019. Named entity recognition on code-switched data: Overview of the calcs 2018 shared task. *arXiv preprint arXiv:1906.04138*.

Kunchukuttan Anoop, Mehta Pratik, and Bhattacharyya Pushpak. 2018. The iit bombay englishhindi parallel corpus. LREC.

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 33–40. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani. 2019. Normalization of indonesian-english code-mixed twitter data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 417–424.

Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2018. Universal dependency parsing for hindi-english code-switching. *arXiv preprint arXiv:1804.05868*.

Eric Brill and Robert C Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th annual meeting on association for computational linguistics*, pages 286–293. Association for Computational Linguistics.

Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(3-4):157–174.

Grzegorz Chrupała. 2014. Normalizing tweets with edit scripts and recurrent neural embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 680–686.

Kenneth W Church and William A Gale. 1991. Probability scoring for spelling correction. *Statistics and Computing*, 1(2):93–103.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Omkar Dhariya, Shrikant Malviya, and Uma Shanker Tiwary. 2017. A hybrid approach for hindi-english machine translation. In *2017 International Conference on Information Networking (ICOIN)*, pages 389–394. IEEE.

Gralinski Filip, Jassem Krzysztof, Wagner Agnieszka, and W Mikolaj. 2006. Text normalization as a special case of machine translation. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pages 51–56.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 368–378. Association for Computational Linguistics.

Mitesh M Khapra, Ananthakrishnan Ramanathan, Anoop Kunchukuttan, Karthik Visweswariah, and Pushpak Bhattacharyya. 2014. When transliteration met crowdsourcing: An empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control. In *LREC*, pages 196–202. Citeseer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1035–1044. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Massimo Lusetti, Tatyana Ruzsics, Anne Göhring, Tanja Samardžić, and Elisabeth Stark. 2018. Encoder-decoder methods for text normalization. Association for Computational Linguistics.

Soumil Mandal, Sainik Kumar Mahata, and Dipankar Das. 2018. Preparing bengali-english code-mixed corpus for sentiment analysis of indian languages. *arXiv preprint arXiv:1803.04000*.

Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. 2019. Neural text normalization with subword units. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 190–196.

Eric Mays, Fred J Damerau, and Robert L Mercer. 1991. Context based spelling correction. *Information Processing & Management*, 27(5):517–522.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. *arXiv preprint arXiv:1809.00388*.

Wookhee Min and Bradford Mott. 2015. Ncsu_sas_wookhee: A deep contextual long-short term memory model for text normalization. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 111–119.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2019. Overview for the second shared task on language identification in code-switched data. *arXiv preprint arXiv:1909.13016*.

Benjamin Muller, Benoît Sagot, and Djamé Seddah. 2019. Enhancing bert for lexical normalization. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 297–306.

Sonja Nießen, Franz Josef Och, Gregor Leusch, Hermann Ney, et al. 2000. An evaluation tool for machine translation: Fast evaluation for mt research. In *LREC*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017. *arXiv preprint arXiv:1803.06745*.

Akshata Phadte and Gaurish Thakkar. 2017. Towards normalising konkani-english code-mixed social media text. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 85–94.

Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. *arXiv preprint arXiv:1611.00472*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ranjan Satapathy, Claudia Guerreiro, Iti Chaturvedi, and Erik Cambria. 2017. Phonetic-based microtext normalization for twitter sentiment analysis. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 407–413. IEEE.

Rajat Singh, Nurendra Choudhary, and Manish Shrivastava. 2018. Automatic normalization of word variations in code-mixed social media text. *arXiv preprint arXiv:1804.00804*.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

Richard Sproat and Navdeep Jaitly. 2016. Rnn approaches to text normalization: A challenge. *arXiv preprint arXiv:1611.00068*.

Rob van der Goot and Gertjan van Noord. 2017. Monoise: Modeling noise using a modular normalization system. *arXiv preprint arXiv:1710.03476*.

Rob van der Goot, Rik van Noord, and Gertjan van Noord. 2018. A taxonomy for in-depth evaluation of normalization for user generated content. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

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.

Hao Zhang, Richard Sproat, Axel H Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. 2019. Neural models of text normalization for speech applications. *Computational Linguistics*, 45(2):293–337.