On Utilizing Constituent Language Resources to Improve Downstream Tasks in Hinglish

Vishwajeet Kumar
IBM Research, India
vishk024@in.ibm.com

Rudra Murthy
IBM Research, India
rmurthyv@in.ibm.com

Tejas Indulal Dhamecha
IBM Research, India
tidhamecha@in.ibm.com

Abstract

Performance of downstream NLP tasks on code-switched Hindi-English (aka *Hinglish*) continues to remain a significant challenge. Intuitively, Hindi and English corpora should aid improve task performance on Hinglish. We show that meta-learning framework can effectively utilize the the labelled resources of the downstream tasks in the constituent[1] languages. The proposed approach improves the performance on downstream tasks on code-switched language. We experiment with *Hinglish* code-switching benchmark GLUECoS and report significant improvements.

1 Introduction

In parts of the world where people speak more than one language in day-to-day affairs, the mixing of the languages often occurs naturally. This phenomenon, termed as code-switching or code-mixing, is observed for various language pairs, e.g. Hindi-English, Spanish-English. Natural language processing tasks such as sentiment analysis, named entity recognition, question-answering are interesting research challenges for *Hinglish* (Hindi-English code-switching).

With the rise of multilingual language models (LM), pretrain-and-finetune approach has been widely used for various downstream NLP tasks; it is observed that when LMs are pretrained with large corpora, they can be easily transferred to downstream tasks with limited fine-tuning data. Most of the publicly available multilingual LMs are pre-trained using combination of various monolingual corpora (such as Hindi and English), but not using code-switched data (e.g. Hinglish). This leads to sub-optimal LMs for the downstream tasks in code-switched language.

Moreover, the downstream tasks on code-switched language (e.g. Hinglish) usually suffers from low-resource problem. Usually, the same task on the constituent languages (e.g. Hindi and English) is not low-resource – which we see as an opportunity. We explore to utilize the constituent language resources to fine-tune the LM for the downstream task in code-switched language. Based on these intuitions, we propose a meta-learning based approach to address the code-switching challenge, as illustrated in Figure. 1. The meta-learner helps leverage Hindi and English resources to improve the performance on code-switched data.

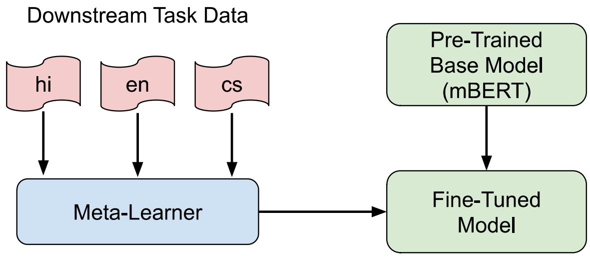

Figure 1: The proposed approach for code-switched NLP utilizing the downstream task data in Hindi and English for improving Hinglish task performance.

During the fine-tuning of a pre-trained language model, its weights adjust for the downstream task. It has been shown that if the base pre-trained model is multilingual, the fine-tuning with one language helps achieve impressive performance on another language in a zero-shot setting too (Pires et al., 2019). It shows that fine-tuning teaches a multilingual model how to perform the downstream task in somewhat language-agnostic fashion. However, such transfers are better when source and target languages are closely related; obviously, the constituent languages are the closest source languages to the target code-switched language. However, the models learned/fine-tuned with constituent languages may become specific to the source languages. Meta-learning approaches are good at solving this problem by learning model parameters suitable for multiple downstream tasks. Therefore,

[1]We use the term *constituent* to jointly refer to the matrix and embedding languages.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3859–3865
December 7-11, 2022 ©2022 Association for Computational Linguistics

we aim to use meta-learning with constituent language resources, to obtain a model state that serves as better initialization point for fine-tuning with limited code-switched samples. Overall, we focus on the following inquiry: *For various downstream tasks, can we effectively use Hindi and English resources to improve the performance on code-switched data?* and explore utility of meta-learning for it.

2 Related Work

Varieties of downstream tasks in code-switched languages have given rise to standard benchmarks and need for synthesizing CS data.

Token-level Language Identification (LID) is one of the earliest explored primary tasks in code-switched NLP for dialectal Arabic-Modern Standard Arab (Elfardy and Diab, 2012; Solorio et al., 2014), Spanish-English, Nepalese-English, Mandarin-English (Solorio et al., 2014), and English-Indic languages (Sequiera et al., 2015) (Zhang et al., 2018).

Part-of-speech (POS) tagging for Hindi-English code-switching has been explored either via LID route (Sequiera et al., 2015) that identified language of text chunks and applied POS tagger of the respective language; or by utilizing language-specific word representations (Ball and Garrette, 2018).

Named Entity Recognition (NER) on code-switched language pairs of Spanish-English, Nepalese-English, Mandarin-English, Modern Standard Arabic-Egyptian (Priyadharshini et al., 2020; Winata et al., 2019; Aguilar et al., 2018; Solorio et al., 2014) have been explored. Use of meta-embeddings (Priyadharshini et al., 2020; Winata et al., 2019) has also been explored for the task of code-switched NER.

GLUECoS and LINCE benchmarks are created by Aguilar et al. (2020) and Khanuja et al. (2020) that include LID, POS-tagging, NER, Question-answering (QA), Sentiment Analysis (SA), Natural Language Inference (NLI), and Machine Translation (MT) for evaluating various models on several language pairs across several tasks.

Code-Switched Text Generation is explored in recent literature to address the data scarcity for code-switched scenarios. Gupta et al. (2020) proposed to create synthetic code-switched texts from parallel corpus by replacing named entities, noun phrases, adjectives in the Hindi sentence with corresponding English translation obtained from alignments. They train a deep learning model on this synthetic data to generate more code-switched texts. Rizvi et al. (2021) have released a toolkit for generating synthetic code-switched texts. They implement two linguistic theories; the Equivalence Constraint theory and the Matrix Language theory to constraint the synthetic code-switched sentences generated. Tarunesh et al. (2021b) use existing unsupervised neural machine translation techniques to generate code-switched sentences. However, in our (limited) initial experimentation with generation approaches, we observe that generated CS text suffers from unnatural switching and grammar violations.

Meta-learning frameworks are task-agnostic representation learning that is widely used for fast-adaptation to downstream tasks of interest. MAML (Finn et al., 2017) and Reptile (Nichol et al., 2018) are arguably two of the most popular meta-learning approaches. MAML's computation and space complexity makes it somewhat impractical – a problem that Reptile addresses with heuristics without compromising on performance. Meta learning is highly effective and shown to be beneficial for a variety of NLP tasks such as low resource machine translation, persona consistent dialogues, low resource sales prediction, speech recognition etc. We believe that its applications can be extended to code-switched NLP too.

3 Our Approach

In this work we incorporate a technique to improve the downstream task performance in Hinglish: by utilize downstream task data of Hindi and English in **meta-learning** framework. Overall, we aim to utilize high resource unsupervised text corpora and task corpora of constituent language (English and/or Hindi) to initialize the model that can be transferred to CS task effectively.

The idea behind meta learning algorithms such as Reptile is to learn a better initialization for the target task using a set of auxiliary source tasks. In CS setting, this amounts to using constituent languages to learn a better initialization parameters and then fine-tuning on small amount of available code-switched data starting from the initialized parameters.

Our specific methodology is inspired from Tarunesh et al. (2021a). To represent it more formally, lets say the code-switched language has English (en) and Hindi (hi) as the constituent lan-

Algorithm 1 Our Meta-learning Approach for CS

Input: Task data T^{en} and T^{hi} for English and Hindi, and T^{hi-en} for Hinglish
Output: The model for Hinglish downstream task

Initialize model weights θ
while not converged **do**
  ▷ Perform Reptile updates
  Draw total of m random batches from T^{en} and T^{hi}
  $\theta_{out} \leftarrow \theta$
  for i^{th} batch, $0 < i < m$ **do**
    train and update model for k steps
    $\theta^{i}_{batch} \leftarrow \theta$
    $\theta \leftarrow \theta_{out}$, reset the model
  end for
  $\theta' \leftarrow \frac{1}{m}\sum_{i=1}^{m}\theta^{i}_{batch}$,
  $\theta_{var} \leftarrow \theta' - \theta_{out}$
  $\theta \leftarrow \theta_{out} + \beta\theta_{var}$,
end while ▷ meta-learning ends.
while not converged **do**
  Fine-tune the model on target T^{hi-en}
end while

guages, lets represent tasks in theses languages as T^{en} and T^{hi} and the task in target code-mixed language as T^{hi-en}. The proposed meta-learning approach is described in Algorithm 1. First, the model weights θ are initialized. The model consists of a randomly initialized classifier head on top of a pre-trained Language Model (LM). The outer loop of the meta-learner executes for n_e number of epochs over T^{hi} and T^{en} task datasets; this is our convergence criteria. Within each meta-step, we sample m batches from the collective pool of batches of English and Hindi. With a batch, the model is trained for k steps, and the updated model weights are persisted as θ^{i}_{batch}, and the model is reset to the θ_{out} state prior to the meta-step. θ^{i}_{batch} represents the *the model state had it been trained with the i^{th} batch alone.* Since each of these states are *too specific*, their average θ' represents *somewhat generalized* state of model for the downstream task. We obtain the measure of the model deviation as θ_{var} that represents the direction of generalized model state θ' relative to the current state θ_{out}. Finally, the model weights are updated in that direction, with step size of β. At convergence, the model state serves as a good initialization to fine-tune the same network on the Higlish task data T^{hi-en}.

4 Experiments and Results

We utilize GLUECoS (Khanuja et al., 2020) benchmark as our test-bed for evaluation on Hindi-English code-switching. The official benchmark involves POS tagging (POS) (Universal Dependency), Named Entity Recognition (NER), Sentiment Analysis (SA), Question Answering (QA) and Natural Language Inference (NLI). We submit test set predictions to the official benchmark portal to obtain the F1-score metrics.

As part of meta-training and its ablations, we utilize the publicly available datasets of each downstream task for Hindi and English languages. The details of the monolingual task datasets used for meta-training are described in Table 1.

4.1 Implementation Details

We use the code repository of Tarunesh et al. (2021a) [2] as base code. We use NVIDIA Tesla V100 GPU to run all experiments. We use `bert-base-multilingual-cased` as base pre-trained model in all our experiments. We tune hyper-parameters based on loss on validation set wherever available. We use the following range of values for selecting the best hyper-parameter

- Batch Size: 8, 16, 32
- Learning Rate: 1e-3, 1e-4, 1e-5, 1e-6, 3e-3, 3e-4, 3e-5, 3e-6, 5e-3, 5e-4, 5e-5, 5e-6

We meta-train the model for $n_e = 5$ epochs, select the best model and fine-tune it further using code-switched data. We set the number of meta update steps $k = 3$, and number of batches in meta-step $m = 8$. We set the meta training step size hyper-parameter $\beta = 1.0$. Meta training requires 12 GPU hours and fine-tuning requires 1 GPU hour, approximately.

4.2 Monolingual Labelled Datasets

Table 1 describes the English and Hindi monolingual datasets used for meta-learning along with their sources and their sizes.

4.3 Baselines and Our Models

To evaluate the effectiveness of our meta training approach, we show comparison with following:

- **CS** that finetunes the base LM using only the CS task data as reported in the leaderboard[3] and we report our replication results too.
- **EN→CS, HI→CS** that first fine-tune base LM on English (in EN→CS) or Hindi (in HI→CS), which is then further fine-tuned for the CS task.

[2] https://github.com/ishan00/meta-learning-for-multi-task-multilingual
[3] https://microsoft.github.io/GLUECoS/

Task	English		Hindi	
	name	#samples	name	#samples
NER	CoNLL 2003	217 K	Hindi NER	60 K
POS UD	UD	217K	Hindi PUD	294 K
SA	Twitter SA	57K	Hindi Tweets	2.7 K
QA	SQuAD v1.1	100 K	SQuAD v1.1 Translated	100 K
NLI	MultiNLI	235 K	Translated-MNLI	263 K

Table 1: English and Hindi downstream task datasets.

- **EN+HI→ CS** that first fine-tunes base LM on combined set of English and Hindi, which is then further fine-tuned for the CS task.
- **EN+HI+CS** training uses concatenated and shuffled English, Hindi and code-switched data. Since there is imbalance in data across three sources, we utilize temperature-based sampling strategy (Arivazhagan et al., 2019) that allows to sample proportional to their dataset sizes ($\tau = 1$) or uniformly ($\tau = \infty$).
- **Meta-Trained** model first uses meta-learning to learn a better model initialization using constituent languages (English and Hindi) and further fine-tunes this meta-trained model using code-switched data.
- **Code-Mixed mBERT:** We also compare our results against the results reported by (Tarunesh et al., 2021b). They further train a mBERT model on synthetic code-mixed sentences generated. The report the results on tasks like NLI and Sentiment analysis.

We use *bert-base-multilingual-cased* (Devlin et al., 2018) as the base model in all our experiments.

4.4 Results and Analysis

Table 2 presents results of our experiments. Following are the key observations:

- Across different tasks, the general trend of performance is observed as Meta-Trained $>$ CS. However, for joint fine-tuning and transfer-learning experiments, we do not observe a clear pattern compared to the baseline model. This indicates the importance of meta-learning in utilizing the task-specific data of the constituent languages in assisting the CS task.
- **Joint fine-tuning** with EN+HI+CS yields somewhat inconsistent results on different tasks when sampling temperature $\tau = 1$. We attribute this to few orders of magnitude of data imbalance between EN, HI, and CS sets. However, enforcing sampling uniformity with $\tau = \infty$ alleviates the problems to some extent. We observe better performance on only Question Answering and Sentiment Analysis tasks.
- **Meta-learning** yields definite improvement across the downstream tasks. It indicates that meta-learning indeed yields a generalized initialization point better suited for fine-tuning with CS.

  Meta-learning and the EN+HI→CS, both, use same fine-tuning data, yet the former yields superior performance. Thus, the improvements by using constituent language data cannot be attributed to the inflated training set *only*.

  Statistical Significance: To understand the statistical significance of the results, we run our replication of CS, transfer-learning experiments, multi-task experiments and the proposed Meta-Trained approach with 5 different random seeds. We perform t-test between the distributions of the obtained performance metrics. We observe that with $p < 0.05$, the proposed Meta-Trained approach outperforms the CS approach, statistically on all the tasks barring NLI task.
- As an auxiliary observation in Table 2, our replication of the mBERT outperforms the one reported on leaderboard and the results reported by Tarunesh et al. (2021b). This indicates towards non-trivial role of hyperparameter tuning on this benchmark.

Based on these observations, the answer to our **RQ** is affirmative. In other words, Hindi and En-

	NER	POS	SA	QA	NLI
CS*	76.96	87.68	57.51	62.23	57.74
CS: Our Replication	76.25 $\pm$ 1.76	88.61 $\pm$ 0.35	59.69 $\pm$ 0.69	62.83 $\pm$ 2.26	59.50 $\pm$ 1.62
Tarunesh et al. (2021b)	-	-	59.39 $\pm$ 0.81	-	59.74 $\pm$ 0.96
EN$\rightarrow$CS	77.22 $\pm$ 1.09	87.50 $\pm$ 1.38	59.40 $\pm$ 0.91	**79.20 $\pm$ 0.04**†	57.48 $\pm$ 0.44
HI$\rightarrow$CS	76.38 $\pm$ 0.60	87.97 $\pm$ 0.38	58.00 $\pm$ 0.77	79.14 $\pm$ 2.15†	56.90 $\pm$ 0.80
EN+HI$\rightarrow$CS	76.83 $\pm$ 0.52	88.23 $\pm$ 0.20	58.90 $\pm$ 0.56	74.15 $\pm$ 3.61†	57.48 $\pm$ 0.50
EN+HI+CS ($\tau = 1$)	69.05 $\pm$ 4.49	87.53 $\pm$ 0.36	**61.96 $\pm$ 0.78**†	58.77 $\pm$ 0.89	49.63 $\pm$ 6.59
EN+HI+CS ($\tau = \infty$)	71.56 $\pm$ 2.08	88.76 $\pm$ 0.42	59.54 $\pm$ 0.61	76.05 $\pm$ 5.34†	56.58 $\pm$ 3.15
Meta-Trained	**78.33 $\pm$ 0.37**†	**89.40 $\pm$ 0.20**†	61.13 $\pm$ 0.68†	78.02 $\pm$ 1.27†	**60.92 $\pm$ 1.11**

Table 2: Results on various GLUECoS tasks. * as reported in the leaderboard. Figures are the F1 metrics. QA uses F1 as defined by squad protocols. Reported figures are mean$\pm$standard deviation. † indicates results which are statistically significant compared to our replication of CS

glish resources help improve the performance on CS task. The improvement is even pronounced when combined with meta-learning approach.

4.5 Qualitative Analysis

We now present few examples from the QA task comparing the predictions from our model and the baseline model.

Example 1
Context: *There is a song by Danish pop group Toy - Box called "Tarzan & Jane", first released as a single in Germany in 1998, and then released worldwide in 1999 to coincide with the release of the Disney film "Tarzan" (see "Film")*
Question: *Tarzan movie kis year me release hui thi ?* (In which year was Tarzan movie released?)
Baseline Prediction: 1998
Our Model: 1999
Ground Truth: 1999

Example 2
Context: *The "Death Note" manga series was first serialized in the Japanese manga magazine "Weekly Shdnen Jump" published by Shueisha in December 2003. . . . In April 2005, "Death Note" was licensed for publication in North America by Viz Media, and the first English language volume was released on October 10, 2005. In February 2008, a one - shot special was released. . .*
Question: *Death note kis desh ka show hai ?* (Death Note show belongs which to country?)
Baseline Prediction: Japanese
Our Model: Japanese manga magazine Weekly Shdnen Jump
Ground Truth: America

In the first example, our model predicts correctly whereas the baseline fails. In Example 2, we observe that both the models fail to predict the correct answer *i.e.,* America and predicts similar incorrect answers.

4.6 Limitations and Future Work

Limitations of this work spawns from 1. requirement of labeled training data in constituent languages and 2. assumption about availability of multilingual pre-trained language model covering both the constituent languages. As we consider the classifier layer weights as part of θ, the universe of labels of constituent languages should encompass the labels for the code-switch data. However, this constraint can be loosened by sharing classifier weights between meta-training and training stages. In future, we would like to validate the proposed approach for other code-switching pairs, such as Spanish-English; in this paper, due to the lack of adequate task-specific datasets in the constituent languages, we were unable to present findings for the ES-EN pair.

5 Conclusion

We studied the usefulness of corpus in the constituent languages, Hindi and English, for improving the performance of the NLP tasks in code-switched language, Hinglish. We propose a Reptile based meta-learning framework which learns better initialization using task-specific labelled datasets in the constituent languages and improves performance for Code-switched language. Our results indicate meta-trained models outperform other strong baselines for all GLUECoS tasks.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2018. Named entity recognition on code-switched data: Overview of the CALCS 2018 shared task. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 138–147, Melbourne, Australia. Association for Computational Linguistics.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

N. Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Z. Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *ArXiv*, abs/1907.05019.

Kelsey Ball and Dan Garrette. 2018. Part-of-speech tagging for code-switched, transliterated texts without explicit language identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3084–3089, Brussels, Belgium. Association for Computational Linguistics.

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*.

Heba Elfardy and Mona Diab. 2012. Token level identification of linguistic code switching. In *Proceedings of COLING 2012: Posters*, pages 287–296, Mumbai, India. The COLING 2012 Organizing Committee.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Hindi NER. `https://github.com/SilentFlame/Named-Entity-Recognition/blob/master/Twitterdata/annotatedData.csv`.

Hindi PUD. `https://github.com/UniversalDependencies/UD_Hindi-PUD`.

Hindi Tweets. `https://github.com/Negibabu/Sentiment-Analysis-of-Hindi-Tweets/`.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Mani Vegupatti, and John P. McCrae. 2020. Named entity recognition for code-mixed indian corpus using meta embedding. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 68–72.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A toolkit for generating synthetic code-mixed text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211, Online. Association for Computational Linguistics.

Royal Sequiera, Monojit Choudhury, Parth Gupta, Paolo Rosso, Shubham Kumar, Somnath Banerjee, Sudip Kumar Naskar, Sivaji Bandyopadhyay, Gokul Chittaranjan, Amitava Das, and Kunal Chakma. 2015. Overview of FIRE-2015 shared task on mixed script information retrieval. In *Post Proceedings of the Workshops at the 7th Forum for Information Retrieval Evaluation, Gandhinagar, India, December 4-6, 2015.*, pages 19–25.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 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, Doha, Qatar. Association for Computational Linguistics.

SQuAD v1.1 Translated. `https://console.cloud.google.com/storage/browser/xtreme_translations`.

Ishan Tarunesh, Sushil Khyalia, Vishwajeet Kumar, Ganesh Ramakrishnan, and Preethi Jyothi. 2021a. Meta-learning for effective multi-task and multilingual modelling. *arXiv preprint arXiv:2101.10368*.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021b. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169, Online. Association for Computational Linguistics.

Translated-MNLI. https://console.cloud.google.com/storage/browser/xtreme_translations/XNLI/translate-train.

Twitter SA. http://alt.qcri.org/semeval2017/task4/data/uploads/download.zip.

UD. https://github.com/UniversalDependencies/UD_English-EWT.

Genta Indra Winata, Zhaojiang Lin, and Pascale Fung. 2019. Learning multilingual meta-embeddings for code-switching named entity recognition. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 181–186, Florence, Italy. Association for Computational Linguistics.

Yuan Zhang, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 328–337, Brussels, Belgium. Association for Computational Linguistics.