Generating Inflectional Errors for Grammatical Error Correction in Hindi

Ankur Sonawane, Sujeet Kumar Vishwakarma,
Bhavana Srivastava, Anil Kumar Singh
Indian Institute of Technology (BHU),Varanasi
{sankur.shrikant.eee16, sujeetkr.vishwakarma.eee16}@itbhu.ac.in
{bhavanasrivastava.rs.cse17, aksingh.cse}@iitbhu.ac.in

Abstract

Automated grammatical error correction has been explored as an important research problem within NLP, with the majority of the work being done on English and similar resource-rich languages. Grammar correction using neural networks is a data-heavy task, with the recent state of the art models requiring datasets with millions of annotated sentences for proper training. It is difficult to find such resources for Indic languages due to their relative lack of digitized content and complex morphology, compared to English. We address this problem by generating a large corpus of artificial inflectional errors for training GEC models. Moreover, to evaluate the performance of models trained on this dataset, we create a corpus of real Hindi errors extracted from Wikipedia edits. Analyzing this dataset with a modified version of the ERRANT error annotation toolkit, we find that inflectional errors are very common in this language. Finally, we produce the initial baseline results using state of the art methods developed for English.

1 Introduction

Grammatical Error Correction (GEC) involves automatically correcting errors in written text, whether relating to orthography, syntax or fluency. Today, most approaches for solving this problem highlight statistical and deep learning methods as opposed to rule-based methods. These methods treat GEC as a translation task, from an ungrammatical to a grammatically correct form of the same language (Brockett et al., 2006). This requires a considerable amount of supervised data in the form of 'edits', which are pairs of incorrect and correct sentences. Researchers have recently done remarkable work on English and a few other resource-rich languages and have released many datasets to evaluate state of the art methods. Comparatively less attention has been given to low resource languages, and Indic languages have been neglected in particular. Systems like UTTAM (Jain et al., 2018) and SCMIL (Etoori et al., 2018) have applied probabilistic approaches and deep learning, respectively, to the problem of spelling correction in Indic languages. Moreover, simple n-gram based models (Singh and Singh, 2019; Kanwar et al., 2017) have been used for "Real-Word" error correction, which is a very similar problem to GEC. However, to our knowledge, no such work exists for true GEC in this language. Thus, we sum up our contributions in the following manner:

1. We create a parallel corpus of synthetic errors by inserting errors into grammatically correct sentences using a rule-based process, focusing specifically on inflectional errors. Since this process is generic, it can easily be extended to other Indic languages.
2. We scrape Hindi edits from Wikipedia and filter them to provide another smaller corpus of errors. Since this corpus is extracted from a relatively natural source, it can be useful for evaluating GEC systems. We also analyze this corpus using an extended version of the ERRANT toolkit.
3. We evaluate a few well studied approaches for languages like English on these datasets, and thus produce the initial GEC results for the Hindi language. The code and data to reproduce our experiments are available at `http://github.com/s-ankur/hindi_grammar_correction`.

2 Related Work

The most common GEC datasets come from correction-annotated language learner essays. The English learner corpora include those from shared

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 165–171

tasks such as Helping Our Own (Dale et al., 2012), CoNLL2014 (Ng et al., 2014) and recently, BEA2019 (Bryant et al., 2019). Similar learner corpora exist for the Russian (Rozovskaya and Roth, 2019) and the Czech (Náplava and Straka, 2019) languages. However, the problem with such manually annotated corpora is that they are not readily available for low resource languages, and their creation will be resource and time-intensive.

Another popular method has been the deliberate injection of errors into grammatically correct sentences, whether by a rule-based system or by strategies like round-trip translation (Lichtarge et al., 2019). The former approach has been essential for languages with limited training data. This was the case for English early on (Izumi et al., 2004; Foster and Andersen, 2009), and is still the case for low resource languages such as Indonesian (Irmawati et al., 2017). Provided that the artificial errors closely resemble real-world mistakes, this method can be applied to obtain large volumes of training data reliably.

A third approach involves mining edits from websites, such as language learner websites (Mizumoto et al., 2011) or from websites with public revision histories like Wikipedia[1] (Grundkiewicz and Junczys-Dowmunt, 2014; Faruqui et al., 2018; Boyd, 2018) or GitHub (Hagiwara and Mita, 2020). While this has the potential to yield natural datasets of considerable size, there are several issues with edits obtained by this method, as not all corrections made in the text are of a grammatical nature; and many simply add more information or are semantic improvements to the text. As the edits lack any human curation, this method results in a more noisy corpus.

3 Hindi Grammar

Hindi is a fusional language that expresses grammatical features like case, gender, number, tense, etc. via morphological changes. In particular, all verbs and some adjectives are inflected to agree with the number and gender of the associated noun (Shapiro, 2003). The same is the case for genitive pronouns, genitive post-positions, and ordinals. Additionally, the verb inflects for the person and the adjective declines for the case of the noun. With a few exceptions, these changes are indicated by vowel endings to the right of the lexical base, as shown with examples in Table 1. If the proper inflection is not specified, then the sentence becomes easily identifiable as ungrammatical due to the loss of agreement.

Gender	Singular	Plural
Masculine	करता karatā	करते karate
Feminine	करती karatī	करती karatī

Table 1: Paradigm for the verb करना (karanā, *"to do"*), showcasing the change in endings according to the gender and number.

4 Error Extraction from Hindi Wikipedia

The WikiEdits 2.0[2] (Grundkiewicz and Junczys-Dowmunt, 2014) software uses Wikipedia revision histories to extract a parallel corpus of errors. We modify this tool for Hindi and extract edits from a Wikipedia revision dump dated October 1, 2020 to create our dataset, which we term as HiWikEd. For filtering the edits, we constrain extracted sentence length to between 6 and 27 tokens and consider only substitution operations with a token-based Levenstein edit distance of less than 0.3. Additionally, we discard edits containing only a difference in punctuation or numbers and corrections involving extremely rare tokens or HTML markups. Edits relating to vandalism are also discarded.

5 Error Analysis

The ERRor ANotation Toolkit (ERRANT[3]) (Bryant et al., 2017; Felice et al., 2016) is a tool that uses morphological and dependency information to analyze, merge and categorize errors using a rule-based system. Initially created for English, it has since been extended to German (Boyd, 2018). We use a similar method to extend the toolkit to Hindi and use it to classify the errors in HiWikEd (See Table 2 for examples). Although the classification criteria consider many exceptional cases, the basic reasoning used by us is as follows:

1. POS tags and lemma for the tokens are obtained using the StanfordNLP tagger (Qi et al., 2018). By comparing POS tags for the edit, the error category is decided as follows.

[1] via http://dumps.wikimedia.org/

[2] http://github.com/snukky/wikiedits

[3] http://github.com/chrisjbryant/errant

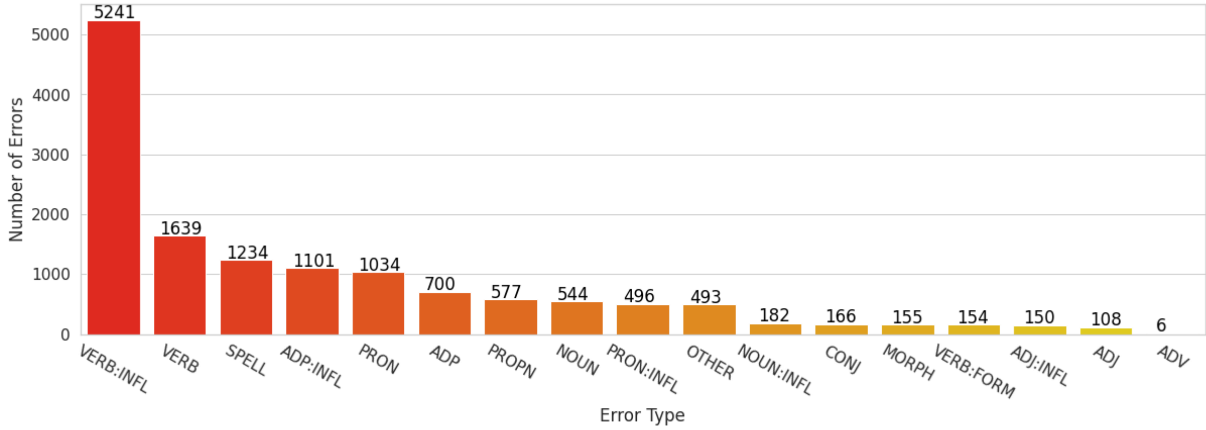

Figure 1: Frequencies of various error types in the HiWikEd dataset.

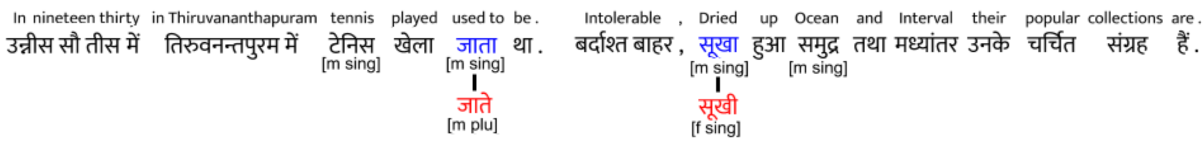

Figure 2: Example of error insertion. In the first sentence, the verb जाता (jātā, *"used to"*) agrees with the noun टेनिस (tenis, *"tennis"*) . We change the inflection of the verb and thus introduce disagreement into the sentence. The same is the case for the adjective सूखा (sūkhā, *"dry"*) in the next sentence.

2. Edits with the same lemma and POS are classified as <POS>:INFL errors and are grammatical in nature. For verbs, an additional category is introduced for tense termed as VERB:FORM.

3. Edits with different lemma but with the same POS are classified as <POS> errors. Most of these are simple semantic changes where one word is swapped for another (e.g. a synonym).

4. Edits having the same stem are classified as MORPH errors.

5. Edits with a low edit distance are classified as SPELL errors while the rest remain unclassified as OTHER.

6 Artificial Error Generation

Since inflectional errors form an easy to identify and common class of Hindi errors, we choose them to generate a synthetic dataset using the following process.

We first extract sentences from the Hindi Wikipedia revision dated June 1, 2020 (using WikiExtractor[4]), assuming that the recent versions are mostly grammatically correct. We tokenize these sentences and POS tag them using the Hindi POS Tagger (Reddy and Sharoff, 2011). We change the inflectional ending for all words of the VERB, ADP, ADV and PRON categories to a different random ending from the inflection table for that POS, taking care that exceptional cases are adequately handled (for examples, refer Table 3). For each of these changes, we create an edit containing a single incorrect word (See Figure 2). We randomly discard 40% of the sentence pairs thus generated. Keeping all sentences generated from a particular correct sentence in the same partition, we split the obtained dataset into train(80%) and valid(20%) partitions (refer Table 6).

7 Experiments and Evaluation

In our experiments, we first test the feasibility of the system using a basic transformer architecture (Vaswani et al., 2018) implemented using the Tensor2Tensor[5] library. For this we use the *transformer_base* setting as a baseline and train the model for 5K epochs. We then evaluate slightly modified versions of two state of the art models relating to English GEC. First, we train the multilayer convolutional encoder-decoder model (Chollampatt and Ng, 2018)[6] for 5 epochs using the de-

[4] http://github.com/attardi/wikiextractor

[5] http://github.com/tensorflow/tensor2tensor

[6] http://github.com/nusnlp/mlconvgec2018

Error Type	Examples
VERB:FORM Verb Tense	बन(ban) → बना(banā), मिलते(milte) → मिलने(milne) make[pres → past], meet[past → inf]
VERB:INFL Verb Inflection	हुआ(huā)→ हुई(huī), रहता(rahatā) → रहते(rahate) happen[m.sing → f.sing], stay[m.sing → m.pl]
NOUN:INFL Noun Inflection	सदस्य(sadasya) → सदस्यों(sadasyon), जिले(jile) → जिला(jilā) member[nom → oblique], district[oblique → nom]
ADP:INFL Postposition Inflection	का(kā) → की(kī), का(kā) → के(ke) of[m.sing → f.sing], of[m.sing → pl]
PRON:INFL Pronoun Inflection	उसका(usakā) → उसकी(usakī), आपने(āpane) → आपको(āpako) his[m.sing → f.sing], you[erg → dat]
ADJ:INFL Adjective Inflection	छोटा(chhotā) → छोटे(chhote), दूसरे(dūsare) → दूसरा(dūsarā) small[m.sing → m.pl], other[m.sing.acc → m.sing]
VERB Verb	रखने(rakhane) → करने(karane), मिला(milā) → दिया(diyā) to keep → to do, found → gave
NOUN Noun	सदी(sadī) → शताब्दी(shatabdī), विश्वास(wishwāsa) → शासन(shāsan) century → centenary, trust → government
ADP Postposition	में(men) → , को(ko), से(se) → का(kā) in → to, from → of[m]
PRON Pronoun	उसके(usake) → उनके(unake), ये(ye) → आप(āp) his → their, these → you
ADJ Adjective	सामान्य(sāmānya) → आम(ām), बड़ा(badā) → छोटा(chhotā) common → ordinary, big → small
ADV Adverb	साथ(sāth) → बाद(bād), बराबर(barābar) → लगातार(lagātār) together → after, correctly → fast
CONJ Conjunction	अगर(agar) → यदि(yadi), पर(par) → परंतु(parantu) if → whether, but → however
MORPH Morphological	बनना(banana) → बनाने(banāne) become → make
SPELL Spelling	कौवा(kauwā) → कौआ (kauā), गई(gaī) → गयी(gayī) crow[spelling], went[spelling]
OTHER Unclassified	और (aur) → के(ke), शहर(shahar) → भी(bhī) and → of[pl], city → and also

Table 2: Error categories of HiWikEd as classified using ERRANT and examples (original → edited).

Error Type	Examples
ADP:INFL Postposition Inflection	की(kī) → के(ke) of[f.sing → pl]
PRON:INFL Pronoun Inflection	मेरा(merā) → मेरी(merī), अपने(apane) → अपनी(apanī) my[m.sing → f.sing], our[m.pl → f.sing]
ADJ:INFL Adjective Inflection	लंबे(lambē) → लंबा(lambā), चौथा(chauthā) → चौथे(chauthē) long[m.pl → m.sing], fourth[m.sing → m.pl]
VERB:INFL Verb Inflection	करता(karatā) → करती(karatī), किये(kiyē) → की(kī) do[m.sing → f.sing], do[pl.past → f.past]

Table 3: List of word categories corrupted using our approach along with examples.

System	ADP:INFL		PRON:INFL		ADJ:INFL		VERB:INFL		Full dataset	
	$F_{0.5}$	GLEU	$F_{0.5}$	GLEU	$F_{0.5}$	GLEU	$F_{0.5}$	GLEU	$F_{0.5}$	GLEU
Transf	0.30	0.62	0.06	0.67	0.06	0.57	0.55	0.79	0.31	0.69
MLConv	0.66	0.81	0.26	0.86	0.36	0.83	0.65	0.83	0.35	0.73
CopyAug	0.70	0.84	0.29	0.71	0.39	0.69	0.70	0.87	0.49	0.80

Table 4: Results for the systems trained on the synthetic corpus and tested on the HiWikEd corpus including the $F_{0.5}$ and GLEU scores. We also specifically report the metrics for the four inflectional categories that we train on.

Source	इस पर उनके पिता राजा नतमस्तक हो गया (gayā). *"on this, his father, the king[m.pl], bowed[m.sing] down"*
Reference	इस पर उनके पिता राजा नतमस्तक हो गये (gaye). *"on this, his father, the king[m.pl], bowed[m.pl] down"*
Output$_{Transf}$	इस पर उनके पिता राजा नतमस्तक हो गया (gayā). *"on this, his father, the king[m.pl], bowed[m.sing] down"*
Output$_{MLConv}$	इस पर उनके पिता राजा नतमस्तक हो गये (gaye). *"on this, his father, the king[m.pl], bowed[m.pl] down"*
Output$_{CopyAug}$	इस पर उनके पिता राजा नतमस्तक हो गये (gaye). *"on this, his father, the king[m.pl], bowed[m.pl] down"*

Table 5: Example system outputs along with source and reference sentences from HiWikEd. The source sentence comes from the Etymology section of Wikipedia article for the Amer Fort.

Dataset	#Sent	#Tok	%Err
Synthetic (Train)	2.6M	45.5M	5.7
Synthetic (Valid)	0.5M	9.1M	5.7
HiWikEd (Test)	13K	208K	6.7

Table 6: Corpus statistics including error percentages, and number of sentences and tokens.

fault hyperparameters. Finally, for training the copy augmented transformer model (Zhao et al., 2019)[7], we skip the pretraining step with the denoising auto-encoder and train the system for 9 epochs.

For model evaluation, we use the GLEU metric (Napoles et al., 2015) as well as the $F_{0.5}$ metric calculated using the Max-Match(M^2) scorer[8](Dahlmeier and Ng, 2012). The systems were trained on the synthetic dataset and then evaluated on the HiWikEd corpus, and the results are presented in Table 4 with an example output shown in Table 5. In addition to the metrics on the full HiWikEd dataset, we calculate the per error type metrics by categorizing the edits using ERRANT, for a more fine-grained analysis of the results.

[7] http://github.com/yuantiku/fairseq-gec
[8] http://github.com/nusnlp/m2scorer

8 Discussion and Future Work

Motivated by the lack of work in GEC for Indic languages, we present two novel error corpora in the Hindi language (as shown in Table 6) and also provide a method for generating a large quantity of artificial inflectional errors. Following error analysis of the HiWikEd corpus using the ERRANT toolkit, we observe that inflectional errors are a reasonably common category in Hindi, making up 49.92% of all errors (see Figure 1).

As seen from the example outputs in Table 5 as well as from the metrics presented in Table 4, the models are able to properly correct many inflectional errors. As expected, the simpler Transformer model is significantly outperformed by the other two models. However, all of the methods perform relatively poorly with regard to the whole dataset, which contains numerous spelling errors and semantic edits for which we do not train our models.

In addition, some grammatical errors in HiWikEd were not inflectional (such as ADJ:FORM) and thus not represented in the synthetic dataset. On manual observation of the dataset, we also find that some edits are identifiably incorrect or simply denote stylistic differences and are out of the scope of GEC. Thus, it may be fruitful to filter and annotate the dataset manually. Including other error types in the training dataset will undoubtedly improve the performance of the model.

Finally, while scraping edits from Wikipedia, we encountered numerous Hindi spelling errors, which we discarded as our focus was solely on grammatical errors. However, these edits may prove to be a valuable source of natural Hindi spelling errors, which can be used to circumvent the dataset problems faced by Etoori et al. (2018) and similar research. Since the approaches used by us for error generation and error categorization are not specific to Hindi, they can easily be extended to other Indic languages like Marathi and Bengali.

References

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Montréal, Canada. Association for Computational Linguistics.

Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152, Melbourne, Australia. Association for Computational Linguistics.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315, Brussels, Belgium. Association for Computational Linguistics.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

Jennifer Foster and Oistein Andersen. 2009. GenERRate: Generating errors for use in grammatical error detection. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 82–90, Boulder, Colorado. Association for Computational Linguistics.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *Advances in Natural Language Processing*, pages 478–490, Cham. Springer International Publishing.

Masato Hagiwara and Masato Mita. 2020. GitHub typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6761–6768, Marseille, France. European Language Resources Association.

Budi Irmawati, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Generating artificial error data for indonesian preposition error corrections. *International Journal of Technology*, 8(3):549–558.

Emi Izumi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2004. Sst speech corpus of japanese learners' english and automatic detection of learners' errors. *ICAME Journal*, 28:31–48.

Amita Jain, Minni Jain, Goonjan Jain, and Devendra K Tayal. 2018. "uttam" an efficient spelling correction system for hindi language based on supervised learning. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(1):1–26.

Shailza Kanwar, Manoj Sachan, and Gurpreet Singh. 2017. N-grams solution for error detection and correction in hindi language. *International Journal of Advanced Research in Computer Science*, 8(7).

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Siva Reddy and Serge Sharoff. 2011. Cross language pos taggers (and other tools) for indian languages: An experiment with kannada using telugu resources. In *Proceedings of the Fifth International Workshop On Cross Lingual Information Access*, pages 11–19, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Michael C Shapiro. 2003. Hindi. *The Indo-Aryan Languages*, pages 250–285.

S. Singh and S. Singh. 2019. Handling real-word errors of hindi language using n-gram and confusion set. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pages 433–438.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, François Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.