Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4299–4305
Marseille, 20-25 June 2022

On the Robustness of Cognate Generation Models

Winston Wu **David Yarowsky**
Center for Language and Speech Processing
Department of Computer Science
Johns Hopkins University
{wswu,yarowsky}@jhu.edu

Abstract

We evaluate two popular neural cognate generation models' robustness to several types of human-plausible noise: deletion, duplication, swapping, and keyboard errors, as well as a new type of error, phonological errors. We find that duplication and phonological substitutions are the least harmful due to cognate training pairs acting as naturally occurring noisy training data, while other types of noise are more harmful. We present an in-depth analysis of the models' performance with respect to each error type to explain the models' predictions.

Keywords: cognates, robustness, seq2seq

1. Introduction

Consider the following scenario: a speaker of a low-resource language, say Ilocano, is searching the web for some information. Because there are relatively few Ilocano web pages, the search engine may perform query expansion by translating the query into a higher-resource language like Tagalog. In many cases, this higher-resource language is a language of wider communication that the user may be familiar with (though may not be fluent in), or is related enough that the user can get the gist of results in that language.

Such a search engine should be robust to the user's query; humans are fallible and often make mistakes. In this work, we examine different types of noise generated by human errors and how these noisy inputs affect the performance of cognate generation models, which can be used to translate the user's query. We investigate typical errors such as typing errors as well as cognitive errors, in which a user not fluent in the language may make spelling errors such as replacing *t* with *d*. These errors include character deletions, duplications, swaps, keyboard errors, and phonologically similar character substitutions (Table 1).

Cognate generation models (Beinborn et al., 2013; Wu and Yarowsky, 2018b; Lewis et al., 2020) translate a single word (e.g. English *tomato*) into a cognate (e.g. Finnish *tomaatti*), a related word that has a common etymological parent (Nahuatl *tomatl*). These models are typically based on neural character-based sequence-to-sequence architectures and have also been increasingly used in a variety of related applications, including language reconstruction (Meloni et al., 2021), modeling etymological and morphological forms (Wu and Yarowsky, 2020a; Vylomova et al., 2020; Wu et al., 2021b), phonology (Wu and Yarowsky, 2021), grapheme-to-phoneme (Gorman et al., 2020), and named entity transliteration (Merhav and Ash, 2018; Wu et al., 2018; Wu and Yarowsky, 2018a). As these neural models become more prevalent in NLP, it is imperative that researchers and practitioners understand how they perform outside of normal conditions.

Error	Input
Original	hispana
Deletion	hispan*d*
Duplication	hi*i*spana
Swap	his*ap*na
Keyboard	hispan*z*
Phonological	hispan*а*

Table 1: Different types of character perturbations, with the misspelled character in red. Notice in the phonological error, the last а is actually a Cyrillic character.

In this paper, we investigate the robustness of two sequence-to-sequence cognate generation models: an LSTM encoder-decoder with attention (Bahdanau et al., 2015) and a Transformer model (Vaswani et al., 2017), which has achieved state-of-the-art on many NLP tasks. We examine the performance of these models on noisy input with errors that are plausibly made by humans. Through in-depth analysis, we show that natural variations in cognate spelling across languages act as training noise that makes the models more robust to certain, but not all, types of errors.

2. Background

There has been a variety of work on cognate prediction and generation. Modern cognate generation models employ a character-based machine translation setup, where the input and output are character sequences of the respective cognates (Beinborn et al., 2013; Wu and Yarowsky, 2018b; Lewis et al., 2020). In addition, successful machine translation models, especially for low-resource languages, have often employed the setup of a single, large multilingual model trained on the concatenation of many languages' data. These systems are generally trained on manually curated cognate lists, or

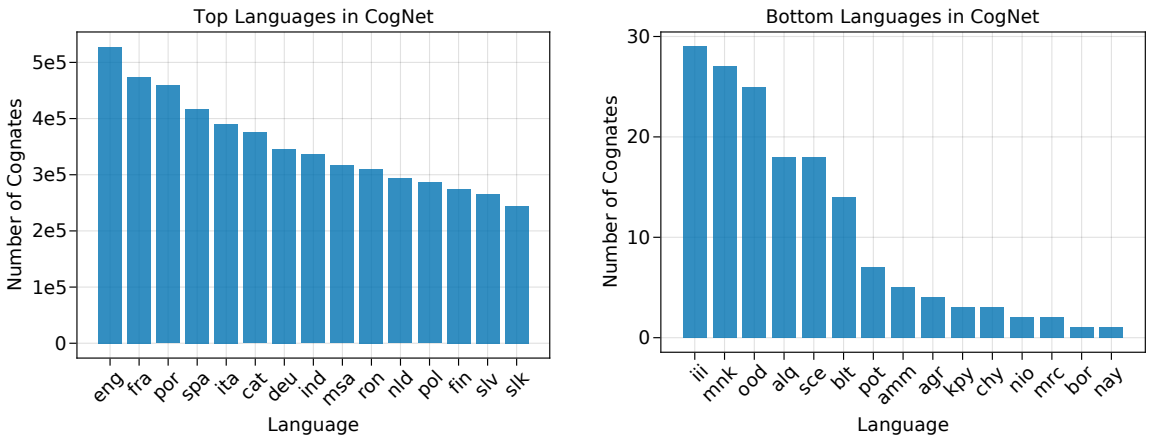

Figure 1: Languages with the most and least data in CogNet 2.0.

in some cases lists automatically extracted from lexical resources such as Wiktionary.

In terms of robustness of systems to misspellings, similar types of noise, also known in the literature as *perturbations*, have been investigated in a variety of NLP tasks, e.g. machine translation (Bertoldi et al., 2010; Formiga and Fonollosa, 2012; Belinkov and Bisk, 2018; Niu et al., 2020; Takase and Kiyono, 2021) and machine comprehension (Jia and Liang, 2017; Wu et al., 2021a). There are existing works that investigate the robustness of systems against adversarial misspellings (Gao et al., 2018; Li et al., 2019; Pruthi et al., 2019). However, in such work, perturbations are usually applied to the entire word, and models are evaluated on whether a change in one word affects the model's understanding of an entire sentence. In contrast, our work focuses on cognate generation, where the input is an single word, broken into characters, and perturbations are applied at the character level.

This type of research into robustness has not yet been of interest to cognate researchers, perhaps because research into cognates is a relatively niche topic. In addition, in the search engine scenario we described above, users are often not maliciously attacking the system but rather might simply accidentally make noisy inputs. Thus, in a real-world application, cognate generations systems should be robust to noise made from non-malicious users. In contrast to previous adversarial work, we also investigate cognitive errors that have not been previously studied.

3. Data

For our experiments, we utilize CogNet 2.0 (Batsuren et al., 2019; Batsuren et al., 2022), a large multilingual database of over 3 million cognate pairs in 338 languages compiled using automated cognate extraction techniques over the Universal Knowledge Core (Giunchiglia et al., 2018).

As with all multilingual datasets, it is important to examine the scope of the languages covered. Figure 1 shows histograms for the most and least common languages in the CogNet database. Perhaps unsurprisingly, high-resource languages such as English, French, and Portuguese have on the order of millions of cognate instances. On the extreme side, languages such as Maricopa (mrc), Bororo (bor), and Ngarrindjeri (nay) have less than 10 instances in CogNet. Thus, the CogNet dataset comprises a wide range of languages around the world, with varying language families, writing scripts, and resourceness.

4. Experiments

To evaluate the robustness of cognate generation models to different types of noise, we select two popular models for cognate generation:

LSTM Encoder-Decoder (Bahdanau et al., 2015) This is a two-layer LSTM encoder-decoder with 512 hidden layer size and 512 embedding size. These are the default settings in OpenNMT-py (Klein et al., 2020).

Transformer (Vaswani et al., 2017) This model has 6 encoder and decoder layers, 8 heads, and 512 embedding size. These are settings recommended by OpenNMT-py.

For our experiments, we split the CogNet database into 80% train, 10% development, and 10% test sets stratified by language, such that each set has the same proportion of words in each language. Because cognacy is a symmetric relation (A is cognate with B implies B is cognate with A), the source-target pairs (A, B) and (B, A) always exist in the same data split in order not to pollute our splits.

Cognate generation is a sequence-to-sequence task, where the input includes the source language, target language, and characters of the word. The inclusion of the target language in the input primes the model to generate the characters of a cognate in the specified target language. The input and output format of the models is as follows:

Input: `eng afr l i n e a r`
Output: `l i n e ê r`

4.1. Noise Models

At test time, we perturb the test data using various noise models described below. We investigate several models of plausible noise based on the scenario of human usage. To test the robustness of the cognate generation systems, we introduce noise by perturbing the spelling of the input word. We distinguish between two types of noise: *Typing Errors*, in which the user knows exactly what they wish to type, but does not hit the correct sequence of keys; and *Cognitive Errors*, in which the user does not know exactly what they want to type and is guessing at the spelling of the word. Specifically, we define the following types of noise.

Deletion (Del) Randomly delete a character. This can be a typing error or a cognitive error.

Duplication (Dup) Randomly duplicate a character. This can be a typing error or a cognitive error.

Swap Swap a random adjacent pair of characters. This is a typing error.

Keyboard (Key) Randomly replace an ASCII character with an adjacent letter on the keyboard. This is a typing error.

Phonological Substitution (Phon) Randomly replace a character with a phonologically similar character or character sequence. This is a cognitive error.

An example of each type of misspelling is shown in Table 1.

4.2. Cognitive Errors

As cognitive errors have not been studied in the context of robustness, we elaborate on how we generate these misspellings. Cognitive errors can be caused by a user who is not fluent in the language. When producing the spelling of an unfamiliar word, the user may draw upon their intuitions of spelling and phonology in their target language as well as their native language. In a similar vein, we generate plausible misspellings by randomly replacing a character with a phonologically similar character or character sequence. This process works as follows.

We first compute character alignments over CogNet cognates pairs by running fast_align (Dyer et al., 2013), a popular word alignment tool. Next, we construct a phrase table using these alignments. Table 2 shows a small portion of this phrase table, along with alignment probabilities. Though the aligner has no notion of phonological distance, it learns that the identity alignment (e.g. *k* aligned to *k*) is the most common alignment, simply because these words are cognates and thus are likely to have sound correspondences. After the identity alignment, the most common aligned characters (or character sequences) are pretty much expected, based on our knowledge of phonology. The aligner learns universal character correspondences such as *k*/*c* which is common across many languages, and can even generalize across characters with diacritics such as *á*/*a*. Notably, since we performed alignment over all 300+ languages *at the same time*, the aligner learns a correspondence between different character sets, e.g. *k* and ᴋ. When we generate phonological substitution misspellings, we randomly replace a letter with one of the top three non-identity aligned character in the learned phrase table.

This process is similar to the calculation of edit weights in a weighted edit distance (Wu and Yarowsky, 2018b), where some string edits have a lower cost because they are close in phonological space (e.g. the substitution of one vowel for another, or the substitution of letters whose sounds have the same place of articulation, like t/d and p/b).

src	tgt	p(tgt | src)
k	k	0.55
k	c	0.22
k	ᴋ	0.07
k	kk	0.02
u	u	0.79
u	o	0.05
u	uu	0.04
u	y	0.03

Table 2: Sample character alignments and their probabilities computed on CogNet.

5. Results

We first examine two evaluation metrics for the task of cognate generation: accuracy and average character edit distance. Figure 2 presents accuracy of the two models over all the error scenarios. We measure 1-best accuracy (is the top model prediction correct) and 10-best accuracy (is the gold in the top 10 model predictions). We find that deletions, character swaps, and keyboard errors lead to the greatest number of model errors, while character duplication and phonological substitutions result in relatively fewer errors. We look at specific examples in the next section.

Our results also show that evaluating these models on 10-best accuracy shows over 2x improvement over 1-best accuracy. In a web search scenario, it is not imperative that a cognate generation model's top prediction is correct, because using these cognates for query expansion, even with misspellings, can lead to higher recall of relevant documents. Unsurprisingly, we find that the Transformer model outperforms the LSTM encoder-decoder model.

We also evaluate on average character edit distance (CED) from the gold, shown in Figure 3. Performance with respect to CED is basically an inversion of accuracy, though the differences are less drastic. Models' performance on the best scenario (no error) and worst scenario (swap) differ on average only by a single character. Thus, higher-performing models do not seem

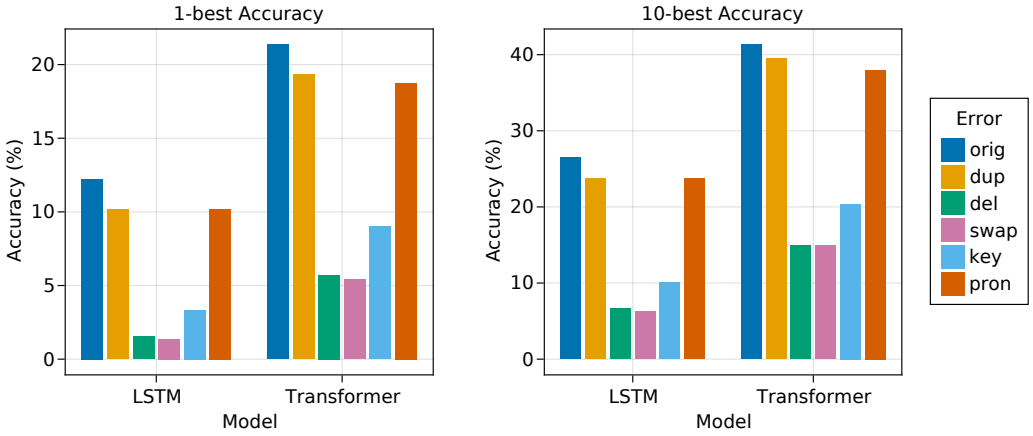

Figure 2: 1-best and 10-best accuracy across models and error scenarios. Higher is better.

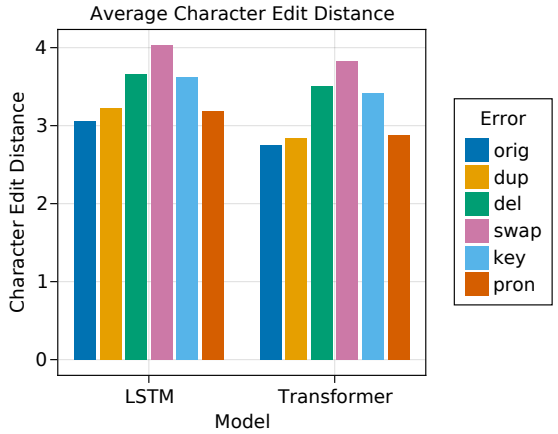

Figure 3: Average character edit distance across models and error scenarios. Lower is better.

to make much of an improvement, on average. After all, cognates tend to have similar spelling. Because of this, we believe that low-performing cognate generation systems can still be useful to allow users to access content in other languages. We would like to evaluate this hypothesis on real users in the future.

6. Analysis

In this section, we look at specific examples to better understand the effects of different types of noise on the performance of cognate generation models. Because the transformer model outperformed the LSTM model in all error types, we analyze the predictions of the Transformer model in the following paragraphs.

One possible interpretation of the contribution of each type of error is that character deletions remove information necessary for generating a cognate in related languages, and character swaps and keyboard errors introduce noise that distorts the spelling of the original word, thus also removing valuable information. In contrast, duplicating and substituting phonologically similar characters do not remove information but rather alter it in some way.

Deletion We just hypothesized that deletion errors remove information necessary to generate cognates in related languages. If we examine the converse, i.e. cases where even with deletions, the model generates the correct cognate, we see that the following characters (in decreasing order of correct predictions) are the least harmful deletions: *e*, *a*, *i*, and *o*. Notably, these are vowels, and it is well known consonants play a more important role than vowels in word recognition (Lee et al., 2001). For example, Vietnamese *uran* → *urn* did not affect the model's correct prediction of ауран. Further down the list, characters whose deletion are most harmful include characters in abugida systems (each character is a syllable) and Chinese characters (each character is an entire morpheme). Thus, deletions indeed remove information, but the amount of information removed is highly dependent on the character that was deleted. Deleting vowels are the least harmful.

Duplication Duplicating characters were one of the least harmful types of noise. The least harmful duplications are: *a*, *i*, *e*, *r*, *n*, *t*, *o*, and *s*. In many languages, duplicating a character only has the effect of lengthening the sound, not changing the place of articulation. And in some cases, duplicating a letter has no effect on the phonology at all. An example is Spanish *español* → *esppañol*, which did not affect the models' predictions. As with deletions, the most harmful duplicated characters come from Sino-Tibetan languages, where modifying a single character is often tantamount to changing an entire word in an alphabetic language (thus duplicating a character is in effect adding an additional word).

Swap Swapping characters overall is detrimental. The least detrimental swapped characters by far are *er*. This is exemplified in *center*/*centre* and *theater*/*theatre*, examples familiar to English speakers from different regions of the world. Following *er*, the least detrimental characters are *ar*, *an*, *me*, and *en*. We notice that /r/, /n/,

and /m/ are all sonorant consonants. Future work can shed further insight into the relationship between the phonology of words and recognition errors.

Keyboard Keyboard errors, in which the user types an adjacent letter on the QWERTY keyboard, are also detrimental to the cognate generation process. The least harmful keyboard errors are replacements with *j*, *s*, and *w*, with the least harmful substitutions being *i* → *j*, *z* → *s*, and *e* → *w*. Some examples of changes that did not affect the model include Afrikaans *reproduktief* → *reproduktjef*, and Adyghe *disember* → *disembwr*. We notice that the least harmful keyboard errors correspond to phonological substitutions.

Phonological Phonological mistakes, caused by cognitive errors when the user is not familiar with the word, are relatively harmless to the cognate generation models. The least harmful substitutions are: *a*, *i*, *e*, *o*, *á*, *u*, and *y*. Some example include Asturian *tribal* → *ᴛribal*,[1] and Turkmen *ýanwar* → *ýanuar*. These misspellings can also include vowel substitutions or accents variations, such as *á* → *a*. The models' robustness to this type of noise can be explained in several ways. A change in script (e.g. *t* → ᴛ) is present in a sufficient amount of training data to be learned by the model. In some cases, such as *w* → *u*, the substitution does not substantially change the word's phonology. The models are also robust to characters missing or adding diacritics. Some languages, such as Spanish, use accents to mark stress; removing this accent also does not drastically affect the word's phonology.

There are also other cognitive processes that can lead to the same mistakes. Some languages, such as French, use diacritics to indicate different sounds, but people may omit accents when typing (e.g. *français* vs *francais*) for a number of reasons, including lack of a French keyboard layout or plain laziness. Even in English, certain borrowed words like *résumé* contain optional diacritics. Other non-standard spellings, such as eye dialect (Wu and Yarowsky, 2020b) or slang, are purposeful misspellings that still preserve most of the phonology of a word. Cognate generation models should be able to handle these types of spelling variation without much difficulty.

7. Conclusion and Future Work

In this paper, we present experiments evaluating several neural cognate generation models' robustness to several types of human-plausible noise. We experiment with several standard noise models (deletion, duplication, swapping, and keyboard errors) as well as a new type of error, phonological errors, finding that the natural variation in cognate spelling across languages have trained the models to be robust to certain types of noise. Duplication and phonological substitution are the least harmful, while character swapping and keyboard errors are more detrimental. We analyzed the models' results with respect to each error type to explain why these models perform as they do. Code for reproducing our experiment is available here.[2]

This study paves the way for future work on robustness in human-AI collaborations. We would like to experiment with more diverse types of noise, perhaps using an adversarial testing framework like TextAttack (Morris et al., 2020) and involving model retraining to improve robustness. Though since humans are generally not adversarial, and we are more interested in the *human* aspect of machine learning, we would like to investigate additional human cognitive errors, such as misspellings made by dyslexic people (Berget and Sandnes, 2015), misspellings in datasets such as the Github Typo Corpus (Hagiwara and Mita, 2020), or perhaps even mispellings by users of alternate keyboard layouts such as Dvorak or AZERTY or different input methods such as Pinyin.

8. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Batsuren, K., Bella, G., and Giunchiglia, F. (2019). CogNet: A large-scale cognate database. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3136–3145, Florence, Italy, July. Association for Computational Linguistics.

Batsuren, K., Bella, G., and Giunchiglia, F. (2022). A large and evolving cognate database. *Language Resources and Evaluation*, 56:165–189.

Beinborn, L., Zesch, T., and Gurevych, I. (2013). Cognate production using character-based machine translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 883–891, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Belinkov, Y. and Bisk, Y. (2018). Synthetic and natural noise both break neural machine translation. *ArXiv*, abs/1711.02173.

Berget, G. and Sandnes, F. (2015). Searching databases without query-building aids: implications for dyslexic users. *Inf. Res.*, 20.

Bertoldi, N., Cettolo, M., and Federico, M. (2010). Statistical machine translation of texts with misspelled words. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 412–419, Los Angeles, California, June. Association for Computational Linguistics.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM

[1] While it may seem somewhat unrealistic to substitute a letter with another letter from a different character set, this is actually not uncommon for bilingual speakers who regularly type in multiple languages. Consider the scenario where the user is in a hurry and simply forgot to switch keyboard input methods.

[2] github.com/wswu/cognate-robust

model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

Formiga, L. and Fonollosa, J. A. R. (2012). Dealing with input noise in statistical machine translation. In *Proceedings of COLING 2012: Posters*, pages 319–328, Mumbai, India, December. The COLING 2012 Organizing Committee.

Gao, J., Lanchantin, J., Soffa, M., and Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

Giunchiglia, F., Batsuren, K., and Freihat, A. A. (2018). One world–seven thousand languages. In *Proceedings 19th International Conference on Computational Linguistics and Intelligent Text Processing, CiCling2018, 18-24 March 2018*.

Gorman, K., Ashby, L. F., Goyzueta, A., McCarthy, A., Wu, S., and You, D. (2020). The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 40–50, Online, July. Association for Computational Linguistics.

Hagiwara, M. and Mita, M. (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, May. European Language Resources Association.

Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September. Association for Computational Linguistics.

Klein, G., Hernandez, F., Nguyen, V., and Senellart, J. (2020). The OpenNMT neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 102–109, Virtual, October. Association for Machine Translation in the Americas.

Lee, H., Rayner, K., and Pollatsek, A. (2001). The relative contribution of consonants and vowels to word identification during reading. *Journal of Memory and Language*, 44:189–205.

Lewis, D., Wu, W., McCarthy, A. D., and Yarowsky, D. (2020). Neural transduction for multilingual lexical translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4373–4384, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Li, J., Ji, S., Du, T., Li, B., and Wang, T. (2019). Textbugger: Generating adversarial text against real-world applications. *ArXiv*, abs/1812.05271.

Meloni, C., Ravfogel, S., and Goldberg, Y. (2021). Ab antiquo: Neural proto-language reconstruction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4460–4473, Online, June. Association for Computational Linguistics.

Merhav, Y. and Ash, S. (2018). Design challenges in named entity transliteration. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 630–640, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. (2020). TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online, October. Association for Computational Linguistics.

Niu, X., Mathur, P., Dinu, G., and Al-Onaizan, Y. (2020). Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8538–8544, Online, July. Association for Computational Linguistics.

Pruthi, D., Dhingra, B., and Lipton, Z. C. (2019). Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy, July. Association for Computational Linguistics.

Takase, S. and Kiyono, S. (2021). Rethinking perturbations in encoder-decoders for fast training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5767–5780, Online, June. Association for Computational Linguistics.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *ArXiv*, abs/1706.03762.

Vylomova, E., White, J., Salesky, E., Mielke, S. J., Wu, S., Ponti, E. M., Hall Maudslay, R., Zmigrod, R., Valvoda, J., Toldova, S., Tyers, F., Klyachko, E., Yegorov, I., Krizhanovsky, N., Czarnowska, P., Nikkarinen, I., Krizhanovsky, A., Pimentel, T., Torroba Hennigen, L., Kirov, C., Nicolai, G., Williams, A., Anastasopoulos, A., Cruz, H., Chodroff, E., Cotterell, R., Silfverberg, M., and Hulden, M. (2020). SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*,

pages 1–39, Online, July. Association for Computational Linguistics.

Wu, W. and Yarowsky, D. (2018a). A comparative study of extremely low-resource transliteration of the world's languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Wu, W. and Yarowsky, D. (2018b). Creating large-scale multilingual cognate tables. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Wu, W. and Yarowsky, D. (2020a). Computational etymology and word emergence. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3252–3259, Marseille, France, May. European Language Resources Association.

Wu, W. and Yarowsky, D. (2020b). Wiktionary normalization of translations and morphological information. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4683–4692, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Wu, W. and Yarowsky, D. (2021). On pronunciations in Wiktionary: Extraction and experiments on multilingual syllabification and stress prediction. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, pages 68–74, Online (Virtual Mode), September. INCOMA Ltd.

Wu, W., Vyas, N., and Yarowsky, D. (2018). Creating a translation matrix of the Bible's names across 591 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Wu, W., Arendt, D., and Volkova, S. (2021a). Evaluating neural model robustness for machine comprehension. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2470–2481, Online, April. Association for Computational Linguistics.

Wu, W., Duh, K., and Yarowsky, D. (2021b). Sequence models for computational etymology of borrowings. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4032–4037, Online, August. Association for Computational Linguistics.