CL-MoNoise: Cross-lingual Lexical Normalization

Rob van der Goot
IT University
robv@itu.dk

Abstract

Social media is notoriously difficult to process for existing natural language processing tools, because of spelling errors, non-standard words, shortenings, non-standard capitalization and punctuation. One method to circumvent these issues is to normalize input data before processing. Most previous work has focused on only one language, which is mostly English. In this paper, we are the first to propose a model for cross-lingual normalization, with which we participate in the WNUT 2021 shared task. To this end, we use MoNoise as a starting point, and make a simple adaptation for cross-lingual application. Our proposed model outperforms the leave-as-is baseline provided by the organizers which copies the input. Furthermore, we explore a completely different model which converts the task to a sequence labeling task. Performance of this second system is low, as it does not take capitalization into account in our implementation. [1]

1 Introduction

Lexical normalization is the task of converting non-canonical text to its canonical equivalent on the word level. As common for Natural Language Processing (NLP) tasks, most of the previous work is done on English data (Han and Baldwin, 2011; Baldwin et al., 2015). However, for lexical normalization there have also been many attempts for other language-(pair)s (Plank et al., 2020; Sidarenka et al., 2013; Alegria et al., 2013; Ljubešić et al., 2017a; Barik et al., 2019; van der Goot et al., 2020; Schuur, 2020; Erjavec et al., 2017; Ljubešić et al., 2017b; Çolakoğlu et al., 2019; van der Goot and Çetinoğlu, 2021), which have been combined into one benchmark for the WNUT 2021 shared task (van der Goot et al., 2021a). Even though data has been available for multiple languages, most work focused on one language, and to the best of our knowledge no one has attempted to solve this task cross-lingually. If successful, a cross-lingual lexical normalization model would open up possibilities for lexical normalization for languages in which no training data is available. In this work, we use the MoNoise model (van der Goot, 2019a) as a starting point, the only normalization model that is open source and has models available for the languages we target. Furthermore, it is heavily dependent on raw data to generate candidates and features, which makes it relatively easy to adapt it for a cross-lingual setup (Section 2). We refer to our new model CL-MoNoise.

In addition to our cross-lingual model, we also evaluate an out-of-the-box sequence labeler, we use the string2string task-type of MaChAmp (van der Goot et al., 2021b), which was originally created for the purpose of lemmatization.

2 Method

2.1 CL-MoNoise

MoNoise is a two-step modular normalization model. It first generates potential normalization candidates, and then ranks these in a second step. For both of these steps a variety of modules are used, and all modules used for generation are also used to generate features for the ranking. The most important candidate generation modules are: the Aspell spell checker[2], closest word in a Twitter word2vec (Mikolov et al., 2013) embedding space, and a lookup list based on the training data. Features from these modules are then complemented by n-gram probabilities based on Wikipedia and Twitter data, Aspell dictionary presence and some language agnostic features, like punctuation detection and length of the candidate (number of characters). For more details on MoNoise, we refer to (van der Goot, 2019a). To retrain MoNoise, new raw data was collected to base its n-gram proba-

[1] https://bitbucket.org/robvanderg/cl-monoise/

[2] www.aspell.net

Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text, pages 510–514
November 11, 2021. ©2021 Association for Computational Linguistics

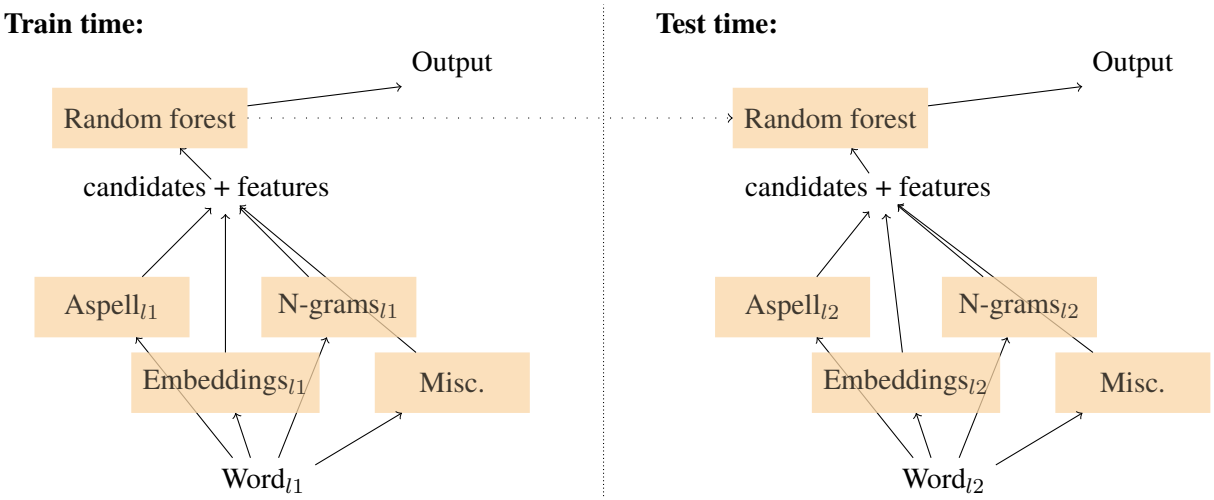

Figure 1: A diagram of our proposed setup to use MoNoise in a cross-lingual setup with source language $l1$ and target language $l2$. All boxes are parts of the model. It should be noted that "Aspell", "N-grams" and "Embeddings" are swapped during test time with target language versions, and the random forest classifier from train time is used. "Misc" here represents all remaining features of MoNoise that can be considered language-agnostic.

bilities and word embeddings on. We downloaded Twitter data of 2012–2020 from archive.org, filtered it with the fasttext language classifier (**?**), and used the most recent Wikidump for each language.[3] This is the exact same data as used by the MoNoise submission provided by the organizers of the shared task (van der Goot et al., 2021a).

We train MoNoise models for each of the source languages, and evaluate them on all the training sets of the other languages to pick the optimal source language for each target language. MoNoise is a supervised model, but many of its features are comprised from language-specific unsupervised components: Aspell spell checker, word embeddings, and n-gram probabilities. We adapt to the new languages by replacing these language-specific modules at run-time. In other words, we train a model on language A, with Aspell, word embeddings and n-gram probabilities based on language A, then we employ this trained model on language B, and use Aspell, word embeddings and n-gram probabilities of language B. This proposed models is No adaptations of the code of MoNoise were necessary, as the data directory is simply a parameter.

We are aware that this setup constitutes an unrealistic setting, as the cross-lingual setup is not pure (annotated data is used to decide which source model to pick). Alternatives could include automatic selection of models based on the input, or to simply use language distances (for example from lang2vec (Littell et al., 2017)). However, we consider this work to be an exploratory analysis, and attempt to validate whether this (cross-lingual) direction is feasible; we leave other strategies for model selection for future work.

We use the Aspell *badspellers* option when it performs better in-language (for all datasets, except SR, ID-EN, SL). For the language pairs (ID-EN, TR-DE), we use the code-switched version of MoNoise (van der Goot and Çetinoğlu, 2021), more specifically the `multi-lingual` model, because we do not assume language labels to be available.

2.2 MaChAmp

As an alternative model, we evaluate the string2string task type of MaChAmp (van der Goot et al., 2021b). This task type uses the Wagner-Fischer algorithm (Wagner and Fischer, 1974) implementation from UDPipe Future (Straka, 2018). This algorithm finds a character edit operation to transform the original word into its normalized form. The training procedure then becomes a sequence labeling problem, where for each word its correct transformation is being predicted. At runtime, the predicted transformation is applied to the original word to obtain the final normalization.

One main weakness of this approach is that it

[3]Of 01-08-2021. Available: https://robvanderg.github.io/blog/twit_embeds.htm

Target	Source DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE
DA	–	2.30	-0.99	-0.59	6.51	-39.71	3.09	-6.38	**8.35**	3.02	-5.06	-0.99
DE	1.01	–	2.44	1.67	0.66	-9.27	10.36	-0.97	-3.96	6.91	12.14	**15.32**
EN	-39.31	1.20	–	-9.34	-45.57	-11.97	**2.74**	-39.38	-41.75	-27.91	-61.85	-50.60
ES	-33.27	-2.53	4.16	–	-32.73	-172.30	**5.06**	-48.82	-67.99	-54.25	-104.90	-19.89
HR	-3.45	-7.03	-3.50	-31.51	–	-106.90	-2.35	-40.34	4.20	**28.54**	-43.25	-36.75
ID-EN	-9.20	5.60	6.91	5.69	-8.47	–	0.54	5.79	-15.82	1.36	**11.68**	7.40
IT	-10.26	-1.62	**2.05**	-5.40	-27.86	-51.84	–	-19.01	-17.93	-20.73	-37.90	-8.53
NL	6.43	17.78	11.03	2.53	6.59	-2.80	15.17	–	8.25	12.64	**19.42**	17.02
SL	1.54	0.90	2.47	-16.90	5.53	-55.32	1.78	-7.93	–	**6.65**	-15.06	-8.71
SR	-2.72	3.26	-4.35	-30.27	**14.77**	-87.43	-0.38	-58.34	-5.88	–	-32.92	-15.81
TR	0.67	9.64	1.22	0.25	4.82	-0.80	5.37	6.25	-2.10	6.00	–	**21.43**
TR-DE	-1.30	8.24	1.91	1.04	-1.88	-2.04	4.05	2.95	-7.01	1.04	**16.54**	–

Table 1: Cross-lingual performance of MoNoise on training splits (ERR).

lowercases all text first, and then tries to predict conversion to capitals where necessary. While this makes sense for lemmatization (its original use-case), this removal of information is probably sub-optimal for lexical normalization, as capitals are often kept.

3 Evaluation

3.1 Development Phase

For the CL-MoNoise model, we tune the source language separately for each dataset. Results are shown in Table 1, most best source languages can be explained by language relatedness, but in some cases the best source languages is surprising; Slovenian scores best for Danish, Turkish is best for Indonesian-English, and Turkish is best for Dutch. We inspected the correct replacements, and found to our surprise that they were not words that exists in both languages, nor were they only some very frequent words. Instead, the word embeddings and Aspell features seemed to have generalized well in spite of the language variety.

For MaChAmp, we use all default hyperparameters, and compare the difference in performance between MBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), which have both been pre-trained on all the languages targeted in MultiLexNorm. Table 2 shows that XLM-R outperforms MBERT for all languages. Furthermore, it becomes clear that capitals are a main weakness of MaChAmp; the uncased scores for DE and NL, which were the only development datasets with capitalization correction in the annotation are much higher. This is because capitalization was not taken into account in the conversion algorithm of MaChAmp, probably because it was focused towards lemmatization, and lemmas are commonly lowercased.

Dataset	cased MBERT	cased XLM-R	uncased MBERT	uncased XLM-R
DE	-90.49	-87.40	30.45	36.94
EN	43.44	48.03	43.44	48.03
HR	34.64	41.83	34.64	41.83
ID-EN	33.68	39.66	33.68	39.66
NL	-25.53	-21.50	17.40	22.45
SL	50.86	57.21	50.86	57.21
SR	35.78	42.76	35.78	42.76

Table 2: Results on all development sets of MultiLexNorm for MaChAmp when using XLM-R or MBERT. Numbers are ERR, with and without taking capitalization into account.

3.2 Test Phase

Results of our two models and all models provided by the organizers are shown in Table 3. MFR is the Most-Frequent-Replacement baseline, which uses the most frequent replacement as found in the target language training data for each token. LAI is the Leave-As-Is baseline, which simply copies over each input word, and thus by definition scores an Error Reduction Rate (van der Goot, 2019b) of 0.0. The results show that results of MaChAmp are dissapointing; even for the languages with no capitalization normalizations it only in the same range as the most frequent baseline (MFR). On the

Model	Avg.	DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE
MoNoise	49.02	51.27	46.96	74.35	45.53	52.63	59.79	21.78	49.53	61.91	59.58	28.21	36.72
MFR	38.37	49.68	32.09	64.93	25.57	36.52	61.17	16.83	37.70	56.71	42.62	14.53	22.09
CL-MoNoise	12.05	7.28	16.55	4.13	4.99	26.41	2.41	0.00	16.22	8.77	20.09	17.57	20.16
LAI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MaChAmp	-21.25	-88.92	-93.36	50.99	25.36	42.62	39.52	-312.87	1.49	56.80	39.44	-12.67	-3.42
Best source lang.		SL	TR-DE	IT	IT	SR	TR	EN	TR	SR	HR	TR-DE	TR

Table 3: Results of the models provided by the organizers (grey) and our proposed models. The source language used for CL-MoNoise is shown in the last row.

first sight, this seems also true for CL-MoNoise, but we should take into account that MFR was trained on in-language training data, whereas CL-MoNoise was not. CL-MoNoise is indeed the best performing system not trained on in-language data, as LAI is the only competitor there; also all other participants in the shared task used in-language training data (van der Goot et al., 2021a). The worst scores for both MaChAmp and CL-MoNoise are on the Italian dataset, which is probably because there are quite some language specific constructions that are normalized (van der Goot et al., 2020), and for MaChAmp it matters that capitalization is corrected. Performance on German, Turkish and the code-switch Turkish-German is relatively high, because they all have highly relevant source languages. For Indonesian-English, performance dropped a lot compared to scores on the development set (Table 1), and it might have been safer to use English as a source language instead of Turkish.

4 Conclusions

In this paper we proposed two models for lexical normalization. 1) CL-MoNoise: based on the original MoNoise; trained with source language raw data and source language annotated normalization data, which is then replaced with target language raw data during run-time. We tuned the source-target language combinations, and found that there are some surprising combinations (SL to DA, TR to ID-EN, TR to NL). Overall results outperform the language-agnostic baseline (LAI), but are outperformed by the most-frequent replacement baseline which is trained on in-language data. 2) MaChAmp: uses a conversion script to turn the task into a sequence labeling task. Performance is especially low for languages that include annotation for capitalization corrections, as this was not taken into account in the MaChAmp implementation. Potential improvements could be made for the MaChAmp system by exploiting its multi-task capabilities, or training a multi-lingual model.

Appendix

I would like to thank the anonymous reviewers for their insightful comments.

References

Inaki Alegria, Nora Aranberri, Víctor Fresno, Pablo Gamallo, Lluis Padró, Inaki San Vicente, Jordi Turmo, and Arkaitz Zubiaga. 2013. Introducción a la tarea compartida Tweet-Norm 2013: Normalización léxica de tuits en Español. In *Tweet-Norm@SEPLN*, pages 1–9.

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, Beijing, China. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

Talha Çolakoğlu, Umut Sulubacak, and Ahmet Cüneyd Tantuğ. 2019. Normalizing non-canonical Turkish texts using machine translation approaches. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 267–272, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tomaž Erjavec, Darja Fišer, Jaka Čibej, Špela Arhar Holdt, Nikola Ljubešić, and Katja Zupan. 2017. CMC training corpus Janes-Tag 2.0. Slovenian language resource repository CLARIN.SI.

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*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Nikola Ljubešić, Tomaž Erjavec, Maja Miličević, and Tanja Samardžić. 2017a. Croatian Twitter training corpus ReLDI-NormTagNER-hr 2.0. Slovenian language resource repository CLARIN.SI.

Nikola Ljubešić, Tomaž Erjavec, Maja Miličević, and Tanja Samardžić. 2017b. Serbian Twitter training corpus ReLDI-NormTagNER-sr 2.0. Slovenian language resource repository CLARIN.SI.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*.

Barbara Plank, Kristian Nørgaard Jensen, and Rob van der Goot. 2020. DaN+: Danish nested named entities and lexical normalization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6649–6662, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Youri Schuur. 2020. Normalization for Dutch for improved pos tagging. Master's thesis, University of Groningen.

Uladzimir Sidarenka, Tatjana Scheffler, and Manfred Stede. 2013. Rule-based normalization of German Twitter messages. In *Proc. of the GSCL Workshop Verarbeitung und Annotation von Sprachdaten aus Genres internetbasierter Kommunikation*.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Rob van der Goot. 2019a. MoNoise: A multi-lingual and easy-to-use lexical normalization tool. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 201–206, Florence, Italy. Association for Computational Linguistics.

Rob van der Goot. 2019b. *Normalization and Parsing Algorithms for Uncertain Input.* Ph.D. thesis, University of Groningen.

Rob van der Goot and Özlem Çetinoğlu. 2021. Lexical normalization for code-switched data and its effect on POS tagging. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics.

Rob van der Goot, Alan Ramponi, Tommaso Caselli, Michele Cafagna, and Lorenzo De Mattei. 2020. Norm it! lexical normalization for Italian and its downstream effects for dependency parsing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6272–6278, Marseille, France. European Language Resources Association.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. 2021a. MultiLexNorm: A shared task on multilingual lexical normalization. In *Proceedings of the 7th Workshop on Noisy User-generated Text (W-NUT 2021)*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021b. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.