Sesame Street to Mount Sinai: BERT-constrained character-level Moses models for multilingual lexical normalization

Yves Scherrer
Department of Digital Humanities
University of Helsinki
yves.scherrer@helsinki.fi

Nikola Ljubešić
Department of Knowledge Technologies
Jožef Stefan Institute
nikola.ljubesic@ijs.si

Abstract

This paper describes the HEL-LJU submissions to the MultiLexNorm shared task on multilingual lexical normalization. Our system is based on a BERT token classification preprocessing step, where for each token the type of the necessary transformation is predicted (none, uppercase, lowercase, capitalize, modify), and a character-level statistical machine translation step where the text is translated from original to normalized given the BERT-predicted transformation constraints. For some languages, depending on the results on development data, the training data was extended by back-translating OpenSubtitles data. In the final ordering of the ten participating teams, the HEL-LJU team has taken the second place, scoring better than the previous state-of-the-art.

1 Introduction

In this paper, we describe the HEL-LJU submission to the MultiLexNorm shared task on multilingual lexical normalization. Lexical normalization is a task of transforming non-standard input tokens into output tokens that follow a specific linguistic standard. In this shared task, lexical normalization is defined even narrower as "the task of transforming an utterance into its standard form, word by word, where one-to-many (1-n) and many-to-one (n-1) replacements are included", disregarding thereby word deletions and insertions (van der Goot et al., 2021). The main motivation behind lexical normalization is to minimize the variability of the linguistic signal, either for computational usage or human consumption. Accordingly, the shared task submissions are evaluated both intrinsically and extrinsically (on a dependency parsing task).

The need for lexical normalization for computational usage is diminishing these days, given the end-to-end methodology that is becoming more and more popular, where the systems are robust enough to accept non-standard input without any explicit normalization. However, for human consumption the need for lexical normalization is still very much present.

The three types of data that still frequently require normalization are user-generated content (i.e., "Internet language" (Ljubešić et al., 2014)), historical data (e.g., 18th century Slovenian (Scherrer and Erjavec, 2016), which is frequently not understood even by native speakers of Slovenian) and dialectal data (very frequently not understood by non-native speakers of a language, or even by the speaker of the same language, as is the case with Swiss dialects of German (Scherrer and Ljubešić, 2016)).

2 Related work

Over the last decade, character-level statistical machine translation (CSMT) has shown very strong results on varying types of non-standard data, such as user-generated content (Ljubešić et al., 2014), historical data (Tjong Kim Sang et al., 2017) and dialectal data (Scherrer and Ljubešić, 2016). Even more, the CSMT approach has shown to behave very similarly in a controlled comparison on various types of non-standard data, such as with Slovenian user-generated content and historical texts (Ljubešic et al., 2016). It has also shown to be the preferred way of adapting language technologies to non-standard data if the availability of human supervision is low (Zupan et al., 2019).

While neural approaches have almost entirely replaced statistical ones in "standard" translation settings (translating between distinct languages), recent studies have shown that SMT-based approaches remain competitive for normalization tasks (Tang et al., 2018; Bollmann, 2019).

Normalization systems not based on translation architectures have also been proposed. For example, MoNoise (van der Goot, 2019) generates a list of normalization candidates for each token and then

Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text, pages 465–472

Code	Language	Words	Sents	Change	Dataset
DA	Danish	16,448	719	9.25%	(Plank et al., 2020)
DE	German	15,006	1,628	17.96%	(Sidarenka et al., 2013)
EN	English	35,216	2,360	6.90%	(Baldwin et al., 2015)
ES	Spanish	7,189	568	7.69%	(Alegria et al., 2013)
HR	Croatian	54,416	4,760	8.89%	(Ljubešić et al., 2017a)
ID-EN	Indonesian-English	13,949	495	12.16%	(Barik et al., 2019)
IT	Italian	12,645	593	7.32%	(van der Goot et al., 2020)
NL	Dutch	12,381	907	28.29%	(Schuur, 2020)
SL	Slovenian	44,944	4,670	15.62%	(Erjavec et al., 2017)
SR	Serbian	56,823	4,138	7.65%	(Ljubešić et al., 2017b)
TR	Turkish	6,443	570	37.02%	(Çolakoğlu et al., 2019)
TR-DE	Turkish-German	12,773	800	24.14%	(van der Goot and Çetinoğlu, 2021)

Table 1: Some statistics on the train-splits of all datasets within the MultiLexNorm benchmark. The 'change' column indicates the percentage of words that are normalized.

ranks them using a variety of features derived from the original text and the proposed normalization candidates.

3 Data

The data in the MultiLexNorm shared task all come from the user-generated-content domain, and comprise mostly of Twitter data. An overview of the 12 datasets is given in Table 1. The sizes of the datasets range between 6 and 56 thousand tokens, with the percentage of changed tokens varying between 7 and 37 percent.

For some of the languages, the data was split into a training and a development set by the organizers. For the other languages, we split the data randomly (90% training, 10% development) and kept the split constant across our experiments.

4 Character-level MT architectures for normalization

Following our earlier experience, we cast normalization as a character-level machine translation problem. In order to enable contextual dependencies, we train and test on entire tweets. We preprocessed the data by replacing URLs by a placeholder and token boundaries by a reserved character. These changes were reverted during postprocessing.

We evaluated an SMT model[1], an RNN-based NMT model[2], and a Transformer-based NMT model[3]. The SMT models are based on Moses (Koehn et al., 2007), the NMT models are based on OpenNMT-py (Klein et al., 2017).

	CSMT	C-RNN	C-TRF
Lg.-specific	92.1%	85.0%	66.7%
Lg.-independent	90.4%	86.4%	89.4%

Table 2: Average normalization accuracies over the validation sets of the 12 languages for different model architectures. Language-specific and language-independent (all training data merged) experiments are reported.

The results are reported in Table 2 (top row) as averages over the 12 languages of the shared task (detailed numbers are given in Table 8 in the appendix). They show a clear advantage for the statistical paradigm (CSMT) over the neural ones (C-RNN and C-TRF). The Transformer-based models were extremely inconsistent across languages, with accuracies below 50% for four languages (DA, ID-EN, NL, TR).[4]

[1]The translation model is monotonous, i.e. without any distortion component. The language model is a character 10-gram model trained with KenLM on the provided training data. The model weights are tuned on the development set with MERT (minimum error rate training) using character error rate as a metric.

[2]The model consists of a bidirectional encoder and a unidirectional decoder, two hidden layers, dimensionality 512 across all layers, and dropout set to 0.1. Maximum sequence length is defined at 1000. We train for a maximum of 50,000 steps with early stopping.

[3]The model consists of 6 Transformer layers with 8 heads each. All dimensionalities are set to 512. Dropout and label smoothing are both set to 0.1. Maximum sequence length is defined at 1000. We train for a maximum of 50,000 steps with early stopping.

[4]We also created CSMT models that were trained and tested on single tokens and thus did not have access to the

	DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE	Avg
No constr.	95.47	88.70	96.80	95.23	95.48	94.82	94.26	80.87	93.52	97.02	76.65	86.26	91.26
BERT	95.23	91.05	97.03	95.30	96.05	94.78	94.85	82.22	93.93	97.44	80.34	87.14	92.11
Oracle	97.03	94.05	97.38	95.89	96.26	95.55	97.64	84.96	94.58	97.62	83.69	92.03	93.89

Table 3: Normalization accuracies with unconstrained and constrained CSMT models.

In order to gauge the potential impact of additional training data, we trained a single language-independent model on the concatenation of all twelve training sets (see bottom row of Table 2 as well as Table 8 in the appendix). For CSMT, this led to lower scores compared to the language-specific models for all twelve languages, whereas the neural models benefitted from the additional data. The scores increased for 9 out of 12 languages with the C-RNN architecture, and for 11 out of 12 languages with the C-TRF architecture, confirming again that NMT models are more data hungry than SMT models.

5 Adding constraints to the normalization process

Table 1 shows that for most of the languages in the task, less than 20% of tokens actually need to be changed. Only three languages in the set require the modification of more than 20% of tokens. Moreover, depending on the language, a substantial proportion of modifications are restricted to casing changes. Hence, the risk of over-normalization, i.e., predicting a spelling change where it is not needed, is significant.

In order to reduce this risk, we propose a sequence labeling task that annotates each token of a tweet with one of five transformation types: *none, capitalize, uppercase, lowercase, modify*. The training data for this task can be directly derived from the MultiLexNorm training data. We then feed these predictions as constraints to the CSMT system, such that only the tokens marked as *modify* are normalized by CSMT. For the remaining categories (*capitalize, uppercase, lowercase*), we transform the tokens via rules and mark them as not to be modified by the CSMT system.

The sequence labeling models are based on pre-trained language-specific BERT or ELECTRA models, which are fine-tuned on the task using derived training data.[5]

Table 3 reports the normalization accuracies of three setups: a CSMT model without any constraints, a CSMT model with constraints predicted by the BERT classifier, and a CSMT model with input constrained by an oracle (the constraints are inferred from the gold annotations of the development sets). The constraints have a positive effect on all languages but Danish and Indonesian–English.[6] In general, the accuracies of the BERT constraints lie about halfway between the unconstrained and the oracle ones.

6 Including synthetic training data from back-translation

The results of the language-independent models of Section 4 suggest that the provided training data is of insufficient size to train reliable translation models, especially neural ones. A well-known strategy to augment the training data in MT is back-translation, where target language data is translated to the source language by an auxiliary model (Sennrich et al., 2016). The resulting parallel data (a standard target side, and a noisy source side) is then included in the training data of the main model.

In the normalization setting, this amounts to finding "clean" data and running it through a model that produces a noisy version of it. To this end, we used filtered subsets of the monolingual OpenSubtitles corpora from OPUS[7] (Tiedemann, 2012) as input data for producing back-translations.

We filtered the OPUS data using the OpusFilter package (Aulamo et al., 2020) and the following filters:

- The length of the line lies between 5 and 25 words (this corresponds to the majority of tweets in the training corpora).

sentential context. As expected, they were consistently outperformed by the models using entire tweets.

[5]We use the NER model class of the *simpletransformers* library (https://simpletransformers.ai/) and fine-tune the models for 10 epochs. The list of pre-trained models is given in Table 7 in the appendix.

[6]The token classification accuracies, i.e. the performance of the BERT classifier before the CSMT normalization step, are provided in Table 9 in the appendix. Classification accuracy seems to be a poor predictor of normalization accuracy though, as illustrated e.g. by the above-average performance of the Danish and Indonesian–English BERT models.

[7]https://opus.nlpl.eu/

	DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE	Avg
Train+Dev	718	2,201	2,950	567	6,348	660	592	1,215	6,227	5,517	569	799	2,364
Full BT	103,559	55,453	88,742	115,713	16,332	85,785	89,295	52,413	58,848	84,162	127,150	75,848	79,442
Sampled BT	11,760	32,560	47,200	8,920		9,900	9,060	18,140		82,760	9,380	12,340	26,433

Table 4: Training instances (tweets/sentences).

	DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE	Avg
No LM/BT	95.47	88.70	***96.80***	*95.23*	95.48	94.82	*94.26*	80.87	93.52	97.02	76.65	86.26	91.26
Full LM	**96.08**	**90.58**	96.70	**95.56**	**95.74**	**95.13**	**94.65**	**82.47**	**94.21**	**97.41**	**79.91**	**86.94**	**92.12**
Full LM+BT	*96.01*	89.59	96.74	94.50	*95.68*	94.74	93.14	*81.78*	*93.93*	*97.28*	*79.57*	*86.80*	91.65
Sampled LM	95.88	90.47	96.75	95.43		94.93	94.36	82.09		**97.41**	78.45	86.56	91.86
Sampled LM+BT	95.84	*89.96*	96.70	94.64		*94.93*	93.80	81.78		97.25	79.23	86.70	91.70

Table 5: Normalization accuracies of unconstrained CSMT models. The LM models include an additional language model trained on the target side of the back-translated data, whereas the LM+BT models additionally include the back-translations for phrase table extraction. Column-wise best results in bold, training setups chosen for the final submissions in italics.

- The line does not contain HTML tags.
- The line only contains Latin script.
- The line is identified as the target language by the *langid* language identifier.[8]
- The line does not contain lower case letters immediately followed by upper case letters (this is an indication of OCR errors or other misspellings).
- The line has a cross-entropy < 20 when evaluated with a language model trained on the training data.[9]
- When normalized with a "forward-translation" CSMT model trained on the training data, the output is identical to the input. This filter is intended to catch typos and non-standard language in the original data, which we want to avoid on the target side.

The resulting dataset is then "unnormalized" using a backward CSMT model trained on the provided training data in the opposite direction, with a beam of 200. Lines whose translation candidates are all identical to the input are rejected. We run the CSMT model for 72 hours per language. Depending on the initial data size, the filters and the speed of the CSMT model, this results in 16k to 127k additional sentences per language (see Table 4).

The resulting back-translations massively outnumber the original training data for most languages, which may affect the final model negatively. Therefore, we also provide random samples of back-translations that contain at most 20 times as many sentence pairs as the given training data (see bottom row of Table 4).[10]

There are two ways of including additional data in an SMT pipeline:

LM Including a second language model trained only on the (non-synthetic) target sides of the back-translated data.

BT Concatenating the original training data with the back-translated data for phrase table extraction.

In addition to the model trained only on the provided data, we therefore obtain four CSMT models, two with the full augmented data and two with the subsampled augmented data. Table 5 shows the results.[11]

We also train C-RNN and C-TRF models with the full set of back-translations, distinguishing the data sources by adding labels on the source side of each sentence. The back-translations increase the results for most languages in all three model

[8] This step is skipped for Serbian because the corresponding *langid* model only matches content in Cyrillic script, whereas the shared task data is entirely in Latin script.

[9] Language model training is also performed within OpusFilter using the default parameters.

[10] We do not provide sampled data for Croatian and Slovene since the total amount of obtained back-translations is lower than the sampling threshold. We did not have the resources to evaluate different sampling thresholds.

[11] Note that the results reported here are without constraints. Also note that the **LM** models were trained as contrastive experiments after the submission deadline and were therefore not considered for the submissions.

	DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE	Avg
Best	68.67	66.22	75.60	59.25	67.74	67.18	47.52	63.58	80.07	74.59	68.58	68.62	67.30
HEL-LJU 2	56.65	59.80	62.05	35.55	56.24	55.33	35.64	45.88	66.97	66.44	51.18	51.18	53.58
HEL-LJU 1	56.65	58.00	60.76	33.68	51.83	53.26	35.64	43.99	66.02	60.26	49.49	51.97	51.80

Table 6: Error reduction rates on the test set. We show the two HEL-LJU submissions and the overall best-performing one.

architectures, but the neural models do not catch up sufficiently to become competitive with the statistical models. The detailed results of the neural models can be found in Table 10 in the Appendix.

For CSMT, it can be seen that the *Full LM* strategy works best overall, but the differences to other setups are small. Since only the *LM+BT* models were available at submission time, we chose the best-performing setup per language among those: the *Full LM+BT* setup for seven languages, the *Sampled LM+BT* setup for two languages and the *No LM/BT* setup for three languages.

7 Submissions

The experiments reported in the previous sections have shown us that for our data and our setup:

- neural character-level MT approaches are not competitive with statistical ones,
- decoding constraints learned with BERT increase normalization accuracies for most languages,
- data augmentation strategies are successful, although the impact of back-translations is negligible in CSMT settings.

For our **first submission**, we choose the CSMT model setup that has led to the best development accuracy for each language (i.e., the setups highlighted in italics in Table 5) and combine it with the BERT-based constraints.

For the **second submission**, we re-create the phrase table and the language model by including the previously held-out development set. We copy the model weights obtained by MERT from the first submission. Note that for those languages where the full back-translations are used, the added development instances amount to a tiny fraction of the overall data. In this case, we expect the results for the two submissions to be very similar.

The results of the intrinsic evaluation are summarized in Table 6. Our submitted systems are ranked 3rd and 4th (out of 18), and we were the second-best team (out of 9). The gap between the best submission and ours is considerable though.

The same ranking is seen in the extrinsic evaluation, although it only concerns German, English and Italian. One should note however that the LAS scores of all systems are very close, showing again that normalization does not provide a substantial advantage for recent downstream-task systems.

8 Conclusion

In this paper we have described our submission to the MultiLexNorm shared task on multilingual lexical normalization. We compare character-level SMT, RNN and Transformer models, showing that in our case, where training data is very limited, SMT still outperforms the two neural options. Increasing the amount of training data by merging data from all languages, or by means of back-translation of OpenSubtitles data, does help the neural approaches, but they still do not perform better than SMT. We further investigate the possibility of predicting via BERT-like models if and how a token should be modified, and show that giving this information to the SMT process improves the final results. Maybe the path to Mount Sinai passes through Sesame Street, after all...

Acknowledgements

This work has been supported by the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 771113), the Academy of Finland through project No. 342859 "CorCoDial – Corpus-based computational dialectology", the Slovenian Research Agency through research core funding No. P6-0411 "Language resources and technologies for Slovene language", the research project ARRS N6-0099 and FWO G070619N "The linguistic landscape of hate speech on social media", and the European Union's Rights, Equality and Citizenship Programme (2014-2020) project IMSyPP (grant no. 875263).

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.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online. Association for Computational Linguistics.

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.

Marcel Bollmann. 2019. A large-scale comparison of historical text normalization systems. 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 3885–3898, Minneapolis, Minnesota. 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.

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.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. 2014. Standardizing tweets with character-level machine translation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 164–175. Springer.

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.

Nikola Ljubešic, Katja Zupan, Darja Fišer, and Tomaz Erjavec. 2016. Normalising Slovene data: historical texts vs. user-generated content. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, volume 16, pages 146–155.

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.

Yves Scherrer and Tomaž Erjavec. 2016. Modernising historical Slovene words. *Natural Language Engineering*, 22(6):881–905.

Yves Scherrer and Nikola Ljubešić. 2016. Automatic normalisation of the Swiss German ArchiMob corpus using character-level machine translation. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS)*.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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*.

Gongbo Tang, Fabienne Cap, Eva Pettersson, and Joakim Nivre. 2018. An evaluation of neural machine translation models on historical spelling normalization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1320–1331, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Erik Tjong Kim Sang, Marcel Bollmann, Remko Boschker, Francisco Casacuberta, FM Dietz, Stefanie Dipper, Miguel Domingo, Rob van der Goot, JM van Koppen, Nikola Ljubešić, et al. 2017. The CLIN27 shared task: Translating historical text to contemporary language for improving automatic linguistic annotation. *Computational Linguistics in the Netherlands Journal*, 7:53–64.

Rob van der Goot. 2019. 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 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: Main Volume*, pages 2352–2365, Online. 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. 2021. 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.

Katja Zupan, Nikola Ljubešić, and Tomaž Erjavec. 2019. How to tag non-standard language: Normalisation versus domain adaptation for Slovene historical and user-generated texts. *Natural Language Engineering*, 25(5):651–674.

Lang.	Type	and model identifier
DA	B	Maltehb/danish-bert-botxo
DE	B	dbmdz/bert-base-german-cased
EN	B	bert-large-cased
ES	B	dccuchile/bert-base-spanish-wwm-cased
HR	E	classla/bcms-bertic
ID-EN	B	bert-large-cased
IT	E	dbmdz/electra-base-italian-xxl-cased-discriminator
NL	B	GroNLP/bert-base-dutch-cased
SL	B	EMBEDDIA/crosloengual-bert
SR	E	classla/bcms-bertic
TR	B	dbmdz/bert-base-turkish-cased
TR-DE	B	dbmdz/bert-base-turkish-cased

Table 7: Pre-trained models used for the token classification task (B = BERT, E = ELECTRA).

		DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE	Avg
CSMT	L-spec	97.88	95.70	96.76	94.89	95.69	94.76	94.33	82.12	93.65	97.13	77.56	84.40	92.1
CSMT	L-ind	96.15	92.63	95.72	93.36	95.26	92.56	92.39	79.32	92.61	97.03	75.24	82.63	90.4
C-RNN	L-spec	92.39	91.44	94.09	81.68	92.72	76.29	88.61	73.84	90.96	94.22	67.46	76.33	85.0
C-RNN	L-ind	81.70	94.07	95.05	91.76	93.31	79.52	92.39	73.76	87.98	95.68	72.18	79.68	86.4
C-TRF	L-spec	41.67	90.73	74.48	75.11	92.70	32.42	89.04	28.89	91.64	93.29	28.50	61.95	66.7
C-TRF	L-ind	96.63	93.85	95.24	92.29	94.40	92.18	93.51	77.32	91.59	95.44	71.50	78.93	89.4

Table 8: Detailed results of the initial character-level MT experiments. *L-spec* refers to language-specific models, *L-ind* to language-independent ones. Note that these experiments use a different train/test split from those reported in Tables 3–6.

	DA	DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE	Avg
BERT	95.57	93.99	97.74	96.75	98.02	96.28	95.28	89.05	96.70	98.69	91.33	90.70	95.01

Table 9: Token classification accuracies.

		DE	EN	ES	HR	ID-EN	IT	NL	SL	SR	TR	TR-DE	Avg
CSMT	No BT	88.70	96.80	95.23	95.48	94.82	94.26	80.87	93.52	97.02	76.65	86.26	90.87
CSMT	Full BT	89.59	96.74	94.50	95.68	94.74	93.14	81.78	93.93	97.28	79.57	86.80	91.25
C-RNN	No BT	84.34	95.22	82.65	93.02	82.70	92.36	76.60	89.92	94.67	68.50	78.28	85.30
C-RNN	Full BT	88.11	94.32	90.73	92.87	47.06	91.63	69.89	90.61	96.54	65.92	79.17	82.44
C-TRF	Full BT	88.56	82.88	75.56	88.14	57.52	73.43	76.29	77.36	82.54	61.55	67.56	75.58

Table 10: Detailed results of the impact of back-translation on different MT architectures. Danish is excluded because those models were trained on a different version of the training data. The average scores of the neural models are negatively influenced by the failing ID-EN models.