Text normalization for low-resource languages: the case of Ligurian

Stefano Lusito[*†]
Universität Innsbruck,
Institut für Romanistik

Edoardo Ferrante[†]
Conseggio pe-o Patrimònio
Linguistico Ligure

Jean Maillard[‡]
Meta AI

Abstract

Text normalization is a crucial technology for low-resource languages which lack rigid spelling conventions or that have undergone multiple spelling reforms. Low-resource text normalization has so far relied upon hand-crafted rules, which are perceived to be more data efficient than neural methods.

In this paper we examine the case of text normalization for Ligurian, an endangered Romance language. We collect 4,394 Ligurian sentences paired with their normalized versions, as well as the first open source monolingual corpus for Ligurian. We show that, in spite of the small amounts of data available, a compact transformer-based model can be trained to achieve very low error rates by the use of backtranslation and appropriate tokenization.

Scintexi

A normalizzaçion da grafia a l'é unna tecnologia d'importansa primmäia pe-e lengue con pöche resorse che mancan de regole fisse à sto reguardo ò ch'en passæ pe varie reforme de scrittua. Fin à oua st'operaçion, pe-e lengue de sta categoria, a l'é stæta basâ ciù che tutto in sce de regole mecaniche, tegnue pe ciù efficaçe che l'utilizzaçion di metodi neurali.

Inte ste pagine analizzemmo o caxo da normalizzaçion pe-o ligure zeneise, unna lengua romansa à reisego de scentâ. Da unna parte emmo arrecuggeito 4.394 frase in ligure e ê emmo accobbiæ co-a seu verscion in grafia normalizzâ; da l'atra, emmo creou o primmo corpus monolingue pe-o ligure à liçensa averta. Into studio demostremmo che, pe quante i dati à dispoxiçion seggian scarsci, l'é poscibile allenâ un modello compatto basou in sciô transformer pe razzonze di basci tasci d'errô, pe mezo da backtranslation e de unna tokenizzaçion appropiâ.

* stefano.lusito@uibk.ac.at.

† Equal contribution.

‡ Work conducted in a personal capacity, independently of the author's affiliation.

1 Introduction

Many recent advances in the field of NLP rely on large-scale textual corpora. Low-resource languages are typically excluded from such developments due to data scarcity issues. To exacerbate the problem, many low-resource languages suffer from noisy data, due to factors such as the absence of rigid spelling conventions or the lack of user-friendly input methods for special characters (Nekoto et al., 2020; Riabi et al., 2021). Text normalization can be a crucial tool to address these issues and enable the creation of corpora from noisy sources (Zupon et al., 2021).

Just as importantly, such technologies can be directly applied to speed up orthographic leveling operations in the field of publishing. Orthographic editing may be desirable when republishing older literary works written according to spelling rules which are no longer in use, or to canonicalize variant forms that may be inadvertently employed even within the same manuscript. This work was born out of such editorial needs for the case of Ligurian, an endangered language spoken within the homonymous region of Liguria in Northern Italy and neighboring territories. The orthographic leveling of historical texts in Ligurian has so far been performed by hand (see e.g. Toso, 1992). The approach presented in this paper represents work conducted by members of the Ligurian linguistic community aiming to largely automate this tedious and time-consuming task with a neural model.

The application of neural methods to text normalization has been limited to high-resource languages (Nguyen and Cavallari, 2020; Wu et al., 2021; Wu and Cotterell, 2019; Tang et al., 2018). Instead, previous work on low-resource text normalization has relied upon hand-crafted rules (Pretorius and Bosch, 2003; Hurskainen, 2004; Asahiah et al., 2017), often perceived as being more data efficient. Indeed the survey by Bollmann (2019)

recommended the use of substitution lists for low-resource languages and suggested reserving statistical and neural approaches for datasets with at least ~10k entries.

In this paper we reassess this belief, looking specifically at the case of Ligurian. We show that, by combining modern data augmentation techniques with a neural model, we are able to achieve average character error rates as low as 2.64 when normalizing texts from a wide range of domains and with a high degree of spelling variation. We achieve this without needing a large-scale annotated corpus: the only resources used are 4.4k unnormalized short sentences (38.1k tokens), manually normalized to produce a parallel corpus (requiring a single native speaker less than five hours of work) and a small unannotated corpus of text (6.7k sentences, 347.5 tokens) which was already in normalized form. We contribute the following:

1. A recipe for training a transformer-based text normalization model for an endangered language, showing how backtranslation (Sennrich et al., 2016a), a commonly applied technique in machine translation, can help boost performance in low-resource scenarios. The model and code are released publicly.[1]
2. A study of the importance of tokenization for neural-based text normalization, showing the effect of varying the vocabulary size for byte-pair encoding (Sennrich et al., 2016b).
3. The creation and public release of the first dataset for Ligurian text normalization as well as the first monolingual corpus for Ligurian.[2]

2 Background

The variety of Ligurian we consider in this paper is Genoese, spoken in the capital and neighboring regions (Toso, 2002). It is not only the most widespread dialect in terms of geographical area and number of speakers, but also the only one to possess a written literary tradition that has developed without interruption from the 13th century to the present day (Toso, 2009).

The spelling system of Genoese has developed over the centuries hand in hand with the evolution of the language itself (Toso, 2009, p. 27-32). A high degree of variation in spelling can still be observed today, due in part to the absence of regulatory bodies and to the lack of keyboard support for several special characters. However, the traditional spelling conventions as formalized by Toso (1997) and recently revised by a group of writers and researchers (Acquarone, 2015) are seeing increasing adoption in the publishing and academic worlds (Toso, 2015; Autelli et al., 2019; Lusito and Maillard, 2021; Lusito, 2022).

In this work, we train models capable of normalizing a variety of Ligurian texts according to the set of spelling conventions mentioned above.

3 Data collection

The texts used for this study were chosen to provide a heterogeneous dataset of variously relevant spellings in Genoese literature. They are:

1. 1,000 sentences and short examples extracted from Casaccia (1876), the most important Genoese-Italian dictionary of the 19th century.
2. Two poems by Piaggio (1846), the most prolific Genoese poet of the first half of the 19th century (1,240 verses).
3. Two issues of *O Balilla* (1,108 sentences in total), one of the main Genoese bi-weekly papers of the 19th and 20th century. Typographical errors were particularly frequent in this newspaper, making it a perfect source of data for this study.
4. Five cantos from Gazzo (1909)'s Genoese translation of Dante's *Comedy* (1,046 verses).

While the first three sources have a similar spelling model[3], Gazzo adopts an extremely complex spelling system, somewhere between the traditional model and a para-phonetic approach (Lusito, 2019, p. 173-175). Unlike the other data sources, Gazzo's spelling system strongly deviates from contemporary Ligurian spelling and does not reflect current usage or even past adoption beyond the author's own work. We nevertheless include it in our analysis to illustrate the model's ability to adapt to strong outliers.

Each one of the datasets above was manually normalized by a native speaker, leading to four parallel corpora which match unnormalized sentences to their normalized versions. For brevity, in experiments we will be referring to these four datasets

[1] https://github.com/fleanend/fairseq-text-normalizer

[2] https://github.com/ConseggioLigure/normalized_ligurian_corpus

[3] See Boano (1997, p. 104-114) for an outline of Casaccia's spelling choices.

as *C*, *P*, *B* and *G* respectively. We use a 70/20/10 training/test/validation split throughout this work.

Additionally, we also use a small monolingual corpus of 6,723 sentences of Ligurian, made up of excerpts from the following sources: *O Stafì*, a contemporary magazine devoted to sociopolitical discussions, a novel (Lusito, 2020), and a dozen articles from Ligurian Wikipedia. These texts are largely already in normalized form, apart from a few simple aspects which were fixed in an automated manner.

The divergences between the different spelling systems of the texts considered in this study are illustrated in figure 1, showing the target normalized form at the top.

Unna	rondaniña	affammâ	a	s’	é	pösâ	in	sciô	teito	de coppi
Ûnn-a	rōndaninn-a	affammâ	a	s’	é	pösâ	in	sciö	teito	de cōppi
Ûnn-a	rondaninn-a	affammâ	a	s’	ë	pösâ	in	sciö	teito	de coppi
Ûnn-a	rondaninn-a	affamâ	a	s’	ë	pösâ	in	sce-o	teito	de coppi
Ûña	rundaniña	affammä’	a	s’	é	pösä	in	sce o	téyto	de cuppi

Figure 1: A sample sentence (*A hungry swallow rests on the tiled roof*) in normalized form (top), compared with how it might have appeared in the unnormalized datasets **C**, **P**, **B** and **G**.

4 Experiments

4.1 Baseline

Our baseline is the transformer architecture used to perform historical text normalization by Wu et al. (2021). After conducting some preliminary experiments to determine if their hyperparameters would be applicable in our highly data scarce setting, we settled upon a much smaller batch size. The resulting architecture, which was trained on `fairseq` (Ott et al., 2019), is made up of 4 encoder and decoder layers with 4 attention heads, embeddings of size 256, a hidden size of 1024, dropout of 0.3 and label smoothing with a factor of 0.1. The model was trained with Adam (Kingma and Ba, 2015) with an inverse square root scheduler, a learning rate of 10^{-3}, 4000 warmup updates, and a batch size of 20.

We train one model per each of the four parallel datasets described in section 3. We then also train one more model on the union of all datasets, with the aim of producing a normalization model capable of working with a wide range of texts.

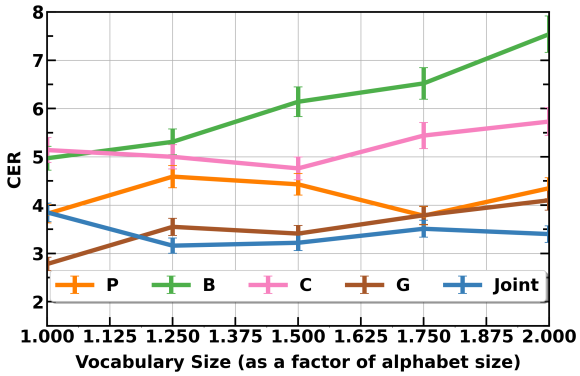

Figure 2: Effect of tokenizer vocabulary size on the validation performance of baseline models. We report the average of three runs and standard error.

4.2 Tokenization

As is common with transformer-based NLP models we apply subword tokenization to our data (Kudo and Richardson, 2018). We hypothesize that vocabulary size might play an important role in a normalization model's performance. We tested several vocabulary sizes, to check whether the tokenization of digraph and initial, medial or caudal position of a grapheme can help or hinder learning. We measure vocabulary size as a factor of the total alphabet size of a dataset, and experiment with values of 1 (corresponding to a character-level model), 1.25, 1.5, 1.75 and 2. Alphabet sizes for the four datasets are 66, 63 , 50 and 78 for the *P*, *B*, *C* and *G* datasets respectively; their union has size 82.

As shown in figure 2, we see that, especially for the spelling systems with the least resources, the level of tokenization (determined by the vocabulary size) has a noticeable impact on the model performance. We suspect the optimal number is highly dependent on the number of unigraph $\leftrightarrow$ multigraph alignment pairs in each parallel dataset: having a multigraph tokenized as a single unit should aid in the task of mapping it to a monograph, while having it split up into multiple tokens should make the model's task harder.

We select vocabulary sizes for each setup based on this validation performance and keep them fixed for the remaining experiments.

4.3 Joint model

A normalization model is most useful if it can operate on texts regardless of their domain or the nature of their spelling variation, and without the need to tag them as being from specific sources. We attempt to create a universal normalizer by merging

	Dataset				
	P	*B*	*C*	*G*	Joint
Specific	3.78	4.97	5.14	**2.78**	4.42
Joint	**2.57**	**3.29**	**2.28**	4.32	**3.16**
Copy	9.54	8.24	21.05	14.59	12.76

Table 1: Test set performance (CER) of the dataset-specific and joint models of section 4.1, compared to a naive "copy" baseline. Best of three runs based on validation performance.

.

	Dataset				
	P	*B*	*C*	*G*	Joint
Specific+BN	2.26	4.95	2.32	**2.39**	3.11
Joint+BN	2.80	2.96	1.67	4.42	2.98
Specific+BN'	2.28	2.45	2.52	2.44	**2.47**
Joint+BN'	**1.91**	**2.38**	**1.36**	4.55	2.64

Table 2: Test set performance (CER) of the dataset-specific and joint models when augmenting training data via "backnormalization" (section 4.4) of 3.7k sentences (+BN) and all 6.7k sentences (+BN') from an unannotated corpus. Best of three runs based on validation performance.

.

all datasets described in section 3 and training one more model on this unified parallel corpus.

Table 1 compares the results of the joint and dataset-specific models. For reference, we additionally report the performance of a naive "copy" baseline, which simply leaves text as-is. We also report the performance on the joint dataset of the dataset-specific models, which consists in sending each sample to the appropriate dataset-specific model depending on its origin.

We find from these results that the joint model still manages to achieve sub-5 CER on all datasets, even outperforming the dataset-specific models in a majority of cases. We believe that concatenating all data helps the model learn the target spelling system better, while consolidating any grapheme normalization rules in common among the sources. On the other hand, the joint model also learns on contrasting evidence for several graphemes, favoring thus the spelling systems with the most features in common (*B* and *C*) and punishing the most eclectic ones. This is particularly evident for the case of *G* which, as already noted in section 3, is by far the most complex to deal with.

4.4 "Backnormalization"

Backtranslation (Sennrich et al., 2016a) is an extremely effective approach used in machine translation to benefit from unannotated, monolingual data (Edunov et al., 2018). We adapt this method to the task of text normalization as follows. We train an additional set of dataset-specific baseline models as in section 4.1, but this time on the reverse task, going from normalized to unnormalized text ("backnormalization" for brevity). We then take our unannotated corpus, which is already in normalized form, and run it through these four backnormalization models. The result of this procedure are four new pseudo-parallel corpora, the target side being our unannotated corpus, and the source side being our backnormalization models' attempts at reconstructing how this text might have been written in unnormalized form, with the spelling variations typical of the four sources of data under study in this paper.

These backnormalized datasets were then used in addition to our original training data. Due to the noisy nature of backnormalized data, which is half-synthetic, in training we upsample the original parallel datasets with a factor of $\lfloor N_{\text{backnormalized}}/N_{\text{original}} \rfloor$ (where N is the length in tokens of a corpus). This is so that in training the model would learn from an equal number of human-normalized and backnormalized data.

To study the effectiveness of backnormalization as well as the effect of the size of the unannotated corpus, we repeat this experiment twice. First using a 3.7k-sized portion of the unannotated corpus, then using the full dataset. The results in table 2 confirm that backnormalization is indeed effective, showing a noticeable reduction in character error rate overall for the models trained on the augmented data. We also a fairly clear trend of improvement in the error rate when more unannotated data is used via backnormalization.

One notable case is performance on dataset *G*, which shows mild improvement in the dataset-specific models but degradation for the joint model. As previously discussed, the spelling system of this dataset represents a clear outlier, deviating strongly from actual contemporary usage. We note that in any kind of real-world use case such type of data would be considered out-of-scope.

5 Conclusion

In this paper we have tackled the issue of text normalization for Ligurian, an endangered low-resource language. We did so by collecting and releasing a dataset of 4,394 Ligurian sentences in different spelling systems paired with normalized versions. We further gathered and released the first open source digital monolingual corpus of contemporary Ligurian, consisting of 6,723 sentences, and showed its potential despite its modest size.

We have shown that in low-resource settings a compact transformer-based model with the appropriate choice of hyperparameters and tokenization, combined with "backnormalization", can achieve CER under 3 points on average, even when the model is given no information on the provenance and spelling conventions of the source text. By varying the size of the corpus used with backnormalization we have further shown that performance is likely to improve even further, should more unannotated data be collected.

There are multiple practical applications of such a model. First of all, it could allow the general public – which tends to experiences difficulties entering Ligurian characters by means of the Italian keyboard – to publish texts in a relatively uniform spelling. Second, on the editorial side, it would greatly speed up orthographic leveling operations, not just of contemporary material but also for the republishing of older literary works written according to spelling rules which are no longer in use. Finally, the normalization of corpora with noisy spelling is especially important for a low-resource language, as it would make more data available for downstream tasks such as language modelling and machine translation which are reliant upon the availability of large corpora.

In conclusion, we consider neural text normalization combined with backnormalization to be a particularly useful tool to promote the preservation as well as the revival of the Ligurian language. The present work only focuses on the case of Ligurian. However, due to the low amount of data needed to train a normalization system, we hope that other researchers and members of language communities in analogous situations will be able to adapt and apply this approach.

References

Andrea Acquarone. 2015. Scrivere la lingua. In Andrea Acquarone, editor, *Parlo Ciæo. La lingua della Liguria*, pages 87–94. De Ferrari / Il Secolo XIX, Genoa, Italy.

Franklin Asahiah, Odejobi Odetunji, and Emmanuel Adagunodo. 2017. Restoring tone-marks in standard yorùbá electronic text: Improved model. *Computer Science*, 18.

Erica Autelli, Konecny Christine, and Stefano Lusito. 2019. GEPHRAS: il primo dizionario combinatorio genovese-italiano online. In Fiorenzo Toso, editor, *Il patrimonio linguistico storico della Liguria: attualità e futuro. Raccolta di Studi*. InSedicesimo, Savona, Italy.

Attilio Giuseppe Boano. 1997. L'alfabeto genovese: dalla codificazione di Giovanni Casaccia alla normalizzazione grafica in atto. *Bollettino dell'Atlante linguistico italiano*, 21:99–133.

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.

Giovanni Casaccia. 1876. *Dizionario genovese-italiano. Seconda edizione accresciuta del doppio e quasi tutta rifatta*. Franchielli, Genoa, Italy.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Angelico Federico Gazzo. 1909. *A Diviña comédia de Dante di Ardighê tradûta in léngua zeneyze cu 'i segni da pronúnçia*. Stampaya da Zoventù, Genoa, Italy.

Arvi Hurskainen. 2004. Swahili language manager: A storehouse for developing multiple computational applications. *Nordic Journal of African Studies*, 13.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Stefano Lusito. 2019. La *Grammatica genovese* di Angelico Federico Gazzo. *Bollettino dell'Atlante linguistico italiano*, 43:155–177.

Stefano Lusito. 2020. Lazarillo de Tormes. In *Cabirda. Lengue e lettiatue romanse*, volume 5. Zona, Genoa, Italy. Translation of an anonymous Spanish novella from 1554.

Stefano Lusito. 2022. *Dizionario italiano-genovese. O diçionäio ch'o mostra o zeneise d'ancheu*. Editrice Programma, Treviso, Italy.

Stefano Lusito and Jean Maillard. 2021. A Universal Dependencies corpus for Ligurian. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Bassey, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Hoang Nguyen and Sandro Cavallari. 2020. Neural multi-task text normalization and sanitization with pointer-generator. In *Proceedings of the First Workshop on Natural Language Interfaces*, pages 37–47, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Piaggio. 1846. *Raccolta delle migliori poesie edite e inedite*. Fratelli Pagano, Genoa, Italy.

Laurette Pretorius and Sonja E Bosch. 2003. Exact hard monotonic attention for character-level transduction. *Southern African Linguistics and Applied Language Studies*, 21.

Arij Riabi, Benoît Sagot, and Djamé Seddah. 2021. Can character-based language models improve downstream task performances in low-resource and noisy language scenarios? In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 423–436, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

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.

Fiorenzo Toso. 1992. Nota al testo. In *Ginn-a de Sampedaenn-a*, pages 24–25. Microart's.

Fiorenzo Toso. 1997. *Grammatica del genovese*. Le Mani, Recco, Italy.

Fiorenzo Toso. 2002. Liguria. In Manlio Cortelazzo, Carla Marcato, and Nicola De Blasi, editors, *I dialetti italiani: storia, struttura, uso*, pages 245–252. UTET, Turin, Italy.

Fiorenzo Toso. 2009. *La letteratura ligure in genovese nei dialetti locali. Profilo storico e antologia*, volume 1. Le Mani, Recco, Italy.

Fiorenzo Toso. 2015. *Piccolo dizionario etimologico ligure. L'origine, la storia e il significato di quattrocento parole a Genova e in Liguria*. Zona, Lavagna, Italy.

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

Andrew Zupon, Evan Crew, and Sandy Ritchie. 2021. Text normalization for low-resource languages of africa. In *AfricaNLP 2021*.