LexiClean: An annotation tool for rapid multi-task lexical normalisation

Tyler Bikaun(✉), **Tim French, Melinda Hodkiewicz, Michael Stewart and Wei Liu**
The University of Western Australia
35 Stirling Highway, Crawley, Western Australia
tyler.bikaun@research.uwa.edu.au
{firstname.lastname}@uwa.edu.au

Abstract

NLP systems are often challenged by difficulties arising from noisy, non-standard, and domain specific corpora. The task of lexical normalisation aims to standardise such corpora, but currently lacks suitable tools to acquire high-quality annotated data to support deep learning based approaches. In this paper, we present **LexiClean**[1], the first open-source web-based annotation tool for multi-task lexical normalisation.

LexiClean's main contribution is support for simultaneous *in situ* token-level modification and annotation that can be rapidly applied corpus wide. We demonstrate the usefulness of our tool through a case study on two sets of noisy corpora derived from the specialised-domain of industrial mining. We show that LexiClean allows for the rapid and efficient development of high-quality parallel corpora. A demo of our system is available at: https://youtu.be/P7_ooKrQPDU.

1 Introduction

Garbage in, garbage out is a well known adage in the computer science and machine learning community. In NLP it has become the centre-focus, demanding a task of its own right; namely, lexical normalisation (Baldwin et al., 2015). Lexical normalisation is the task of identifying and normalising non-canonical tokens (e.g. erroneous spelling, acronyms, ...) in noisy, non-standard, corpora (Han and Baldwin, 2011).

Largely made popular after the 2015 ACL-IJCNLP Workshop on Noisy User-generated Text (W-NUT) (Baldwin et al., 2015), lexical normalisation has demonstrated marked improvements on down-stream applications such as entity recognition, text classification, and part-of-speech (POS) tagging (Derczynski et al., 2013; Hua et al., 2015; Van der Goot et al., 2017; Núñez et al., 2019). These improvements have centred around the fact that many NLP tools are not amenable to noisy corpora, such as those in micro-blogging domains like Twitter (Liu et al., 2011), and in specialised-domains such as industrial mining (Stewart et al., 2018).

To date the most popular lexical normalisation corpus is based on English Twitter and was released as part of W-NUT (Baldwin et al., 2015). This has resulted in a number of algorithmic contributions to lexical normalisation task with the current state-of-the-art using ensemble learning methods (van der Goot and van Noord, 2017). More recently, attention has shifted towards neural techniques that i) contextually normalise tokens based on high-level classifications (Stewart et al., 2019b), ii) modify and fine-tune large pre-trained transformer based representations (Muller et al., 2019), or iii) perform joint normalisation and sanitisation (e.g. masking sensitive tokens) (Nguyen and Cavallari, 2020).

However, neural models typically demand large volumes of high-quality training data, which is not available for the task of lexical normalisation. Despite the prevalence of open-source token-level annotation tools (Stenetorp et al., 2012; Yimam et al., 2013; Yang et al., 2017; Kummerfeld, 2019), there still remains a lack of support for lexical normalisation.

A gap in lexical normalisation research currently exists and consists of an absence of large scale annotated corpora and scalable, task-specific tools for their construction. To fill this gap, we introduce **LexiClean**, an annotation tool for multi-task lexical normalisation that is:

i. *Rapid*: Enables fast corpus wide multi-task annotation.

ii. *Flexible*: Supports *1:1* and *1:N* token normalisation.

iii. *Intuitive*: Maintains a simple and easy-to-use interface.

[1] *LexiClean*. https://lexiclean.nlp-tlp.org

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 212–219
November 7–11, 2021.

iv. *Dynamic*: Permits organic schema development during annotation.

The remainder of this paper is organised as follows. We define the task of lexical normalisation in Section 2 and briefly review related work in Section 3. Following this, we present and describe key features of LexiClean in Section 4. LexiClean's system architecture is then discussed in Section 5 with a case study presented in Section 6. Lastly conclusions are drawn and future work is proposed in Section 7. An online demonstration of LexiClean is located at `https://lexiclean.nlp-tlp.org` and the source code is available under an Apache-2.0 license at `https://github.com/nlp-tlp/lexiclean`.

2 Problem Formulation

Lexical normalisation is defined as the mapping of non-canonical, out-of-vocabulary (OOV) tokens to canonical, in-vocabulary (IV) forms (Han and Baldwin, 2011). Non-canonical tokens are largely a result of i) unconventional and phonetic spelling, ii) improper casing, iii) acronyms, iv) abbreviations and initialisms, v) domain-specific terms, vi) neologisms, and vii) erroneous concatenation or tokenization. This task is akin to grammatical error correction (GEC) (Ng et al., 2014), although it does not involve token reordering that is core to GEC.

Lexical normalisation is typically tackled as one of two formulations, either as a sequence-to-sequence (seq2seq) (Muller et al., 2019; Nguyen and Cavallari, 2020) or token classification problem (van der Goot and van Noord, 2017; Stewart et al., 2018, 2019b). Seq2seq structures the learning task similar to neural machine translation (NMT) (Bahdanau et al., 2014) whereby an *encoder* receives a sequence of noisy text, $\mathbf{X} = (x_1, \ldots, x_n)$, and maps it to a *decoder* which outputs a sequence of normalised text, $\mathbf{Y} = (y_1, \ldots, y_m)$. In this format, $|\mathbf{X}|$ does not necessarily have to equal $|\mathbf{Y}|$. Here a variation in sequence length can result from concatenation and tokenization corrections e.g. $(\text{“}helloworld\text{”}) \rightarrow (\text{“}hello\text{”}, \text{“}world\text{”})$ or $(\text{“}hello\text{”}, \text{“}w\text{”}, \text{“}orld\text{”}) \rightarrow (\text{“}hello\text{”}, \text{“}world\text{”})$.

In contrast, token classification structures the task in a modular fashion where OOV candidates are *identified* and *normalised* in multiple stages. Typically a noisy sequence, $\mathbf{X}$, is mapped to an intermediate sequence of semantic classes, $\mathbf{Z} = (z_1, \ldots, z_n)$. Token classification can be simple binary classification, $\mathcal{L}_{n=2} = \{OOV, IV\}$, or comprehensive, $\mathcal{L}_{n=4} = \{self, spelling\_error, domain\_specific, acronym\}$, where $\mathcal{L}$ is a space consisting of n pre-defined classes of token categories. After classification, alignment to suitable canonical forms is performed using similarity or distance based measures conditioned on labels in $\mathbf{Z}$ (Han and Baldwin, 2011; Baldwin et al., 2015).

3 Related Work

In the last decade, many open-source annotation tools have been developed for token-level classification tasks such as entity recognition and POS tagging, notably BRAT (Stenetorp et al., 2012), WebAnno (Yimam et al., 2013), YEDDA (Yang et al., 2017), and SLATE (Kummerfeld, 2019). The contributions of the current generation of tools have been significant, but support for the task of lexical normalisation has been overlooked. As a result, these tools do not have features that enable *in situ* token modification or data quality improvements such as decatentation and tokenization whilst performing their main tasks. On the other hand, proprietary writing assistants such as Grammarly[2], ProWritingAid[3], and Ginger[4] do contain features required for lexical normalisation, but are prohibitively expensive and not designed for the task of corpora annotation.

4 LexiClean - Key Features

This section provides an overview of the key features of LexiClean that enable rapid multi-task token-level annotation that supports both seq2seq and token classification task formats. An overview of the system is presented in Figure 1 with a web-based interface in Figure 3.

4.1 Project Creation and Automatic Labelling

LexiClean provides users upon project creation the facility to upload a predefined OOV to IV (*1:1*) replacement dictionary (e.g. *{"hel" : "hello", "worl": "world"}*) and an unlimited number of plain-text gazetteers (Figure 3). Gazetteers are lists of tokens mapped to a high-level concept (e.g. *domain_specific → {u/s, . . . , c/o}*. Here, these concepts are referred to as *meta-tags* and are used to support the token classification formulation of lexical normalisation. These resources are used to

[2]Grammarly. https://www.grammarly.com/
[3]ProWritingAid. https://prowritingaid.com/
[4]Ginger. https://www.gingersoftware.com/

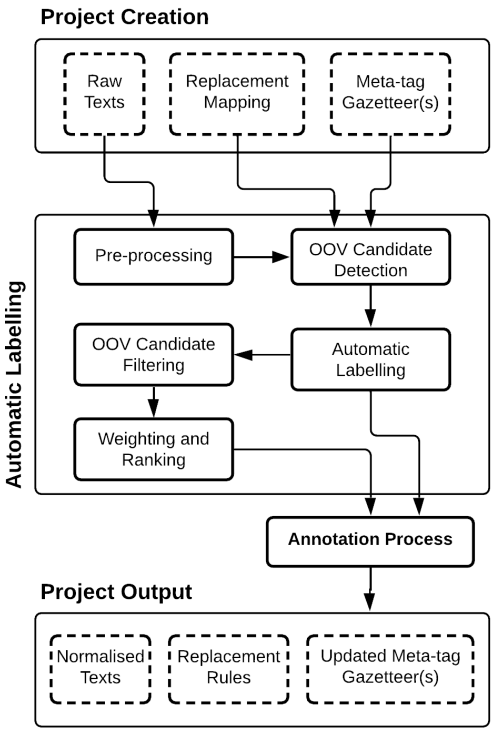

Figure 1: Overview of LexiClean process and data flow.

automatically label tokens in the entire corpus before an annotation session commences (Figure 1), notably reducing annotation effort.

Depending on the resources used, replacements will be automatically applied as *suggested replacements* (Figure 3(a)) whereas meta-tags will applied directly (Figure 2). However, any accepted suggested replacements or automatically applied meta-tags can be removed at any time throughout an annotation session if deemed unsuitable (see Figure 3(b) and Figure 2).

4.2 Single and Multiple Replacements

Instead of iteratively constructing replacement dictionaries only as a *1:1* mapping throughout the annotation process, LexiClean allows the correction of single tokens *in situ* (*1:1*) or across the entire corpora via *cascading* (*1:N*) (see *apply* and *apply all* in Figure 3(c)).

This has two main benefits: i) single non-canonical tokens can be replaced *in situ* enabling contextual normalisations to be captured, and ii) cascading replacements across the entire corpora hastens annotation speed. The importance of this is illustrated by considering the following texts - *around the wod*, *cut the wod*, and *burn fire wod*. *1:1* dictionary based methods (e.g. replace all) would only be able to capture the replacement as either *wood* or *world* which would incorrectly annotate either 1 or 2 of the texts. Here, LexiClean allows users to modify *wod* $\rightarrow$ *world in situ* and cascade *wod* $\rightarrow$ *wood* across the remainder of the corpus (if deemed suitable). In some instances, the application of both styles of normalisation can indirectly lead to *N:1* mappings being formed.

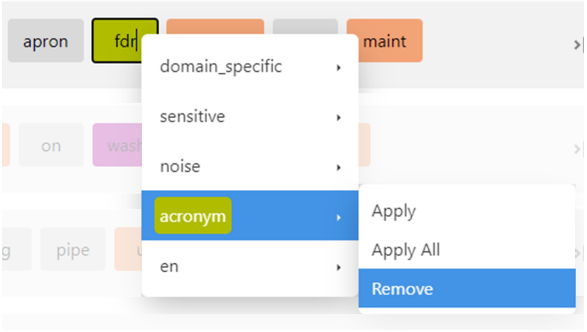

Figure 2: LexiClean meta-tag context menu.

4.3 Easily Identifiable Token Markup

Identifying and normalising OOV tokens in large corpora can be a demanding task, especially over thousands of texts. As a result consistency can be negatively impacted due to the inability of a user to recall corrections they have made to non-canonical token forms. To overcome this, LexiClean marks up tokens using a colour system. Colours for replacements, suggested replacements, and IV and OOV candidates are set to a default palette (Figure 3) whereas meta-tag colours are specified by the project creator on project creation. By using distinct colours to markup tokens, rapid identification can be ensured and consistency preserved. For example, users can quickly see where suggestions have been made and decide to *accept* or *ignore* them.

4.4 Dynamic Schema

Similar to token-level annotation tools that employ dynamic schemas (Stewart et al., 2019a), LexiClean allows users to update their meta-tag schema throughout the annotation process. This feature permits users to organically modify their schema based on phenomena present in the corpora rather than fitting to a prescriptive set of classes. Updates include additional classes of meta-tags and toggling the *active* state of existing ones. Toggling of meta-tag active states within the schema permits

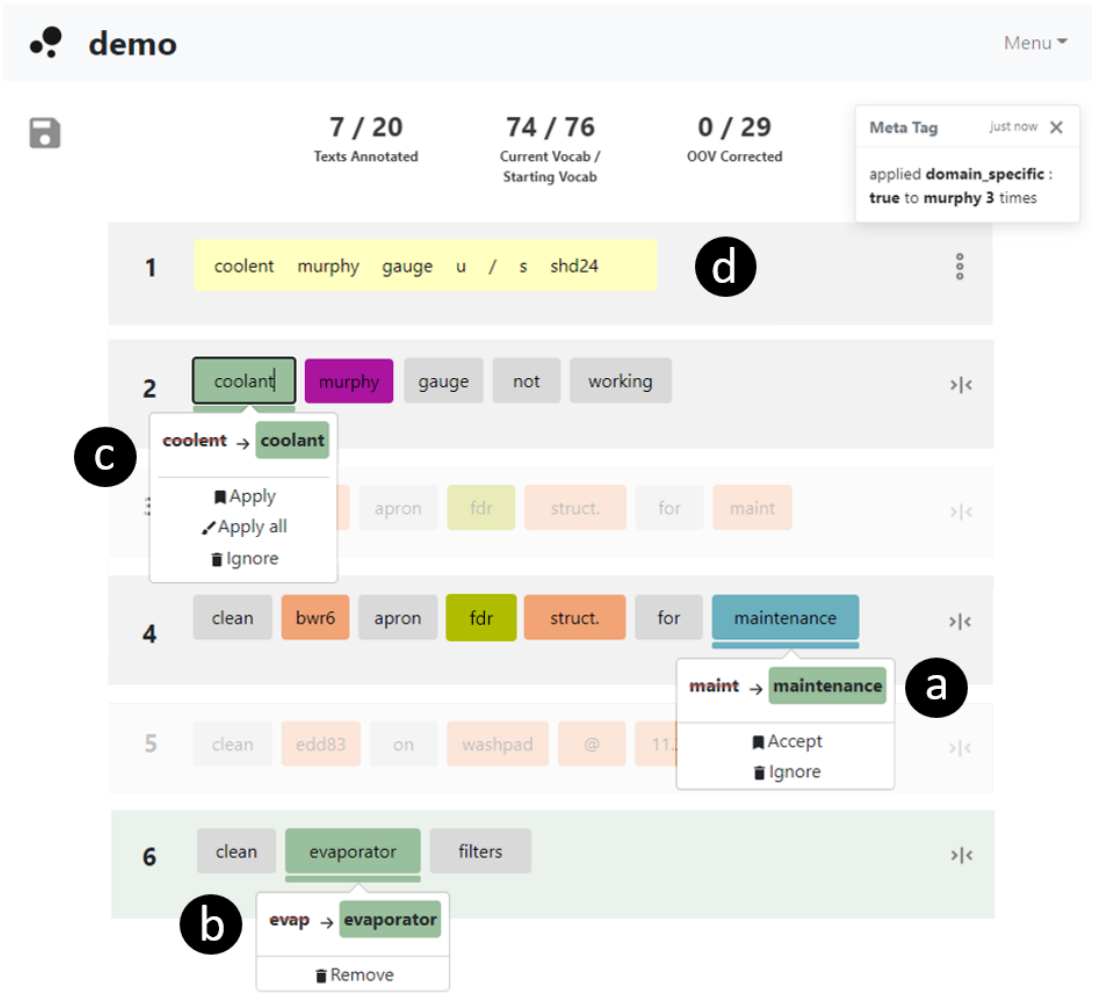

Figure 3: LexiClean annotation interface - (a) suggested token replacement, (b) accepted token replacement, (c) in-progress token normalisation, and (d) text tokenization mode.

a soft-deletion that can be reversed if required by the user.

4.5 Decatenation and Tokenization

Concatenation and irregular tokenization of texts are common in noisy corpora. Consider the following problematic example that exhibits both cases:

original	hewalkedacross th er oad
corrections	he, ␣, walked, ␣, across, {th er → the}, ␣, {r oad → road}
normalisation	he walked across the road

LexiClean manages this by first allowing the user to decatenate the concatenated tokens by introducing additional white space (␣). Secondly, incorrect tokenization is corrected through a utility function that allows users to change the annotation mode of a text and modify its token spans (see Figure 3(d)).

4.6 Sorting Algorithm

To optimise annotation speed, LexiClean computes the average inverse tf-idf weight (Manning and Schutze, 1999) on project creation from all OOV candidates in each text. Using these weights, texts are presented to the user in ranked order with the most prominent candidates appearing first. The rationale behind this technique is that the immediate annotation of high-frequency OOV candidates will have a significant impact on the conversion rate of texts when using the *cascade* style annotation.

4.7 Exporting Annotations and Normalisation Maps

At any stage of an annotation project, users can download their annotated corpora in an extended W-NUT JSON-based format (Baldwin et al., 2015). Additionally, replacements and meta-tag gazetteers generated over the course of the project can also be exported for use in new projects or external

systems.

5 System Architecture

LexiClean is built using the modern full stack web development framework *MERN*[5] (MongoDB-Express-React-NodeJS). All annotations are captured at the token-level as shown in the MongoDB (NoSQL) entity relationship diagram in Figure 4.

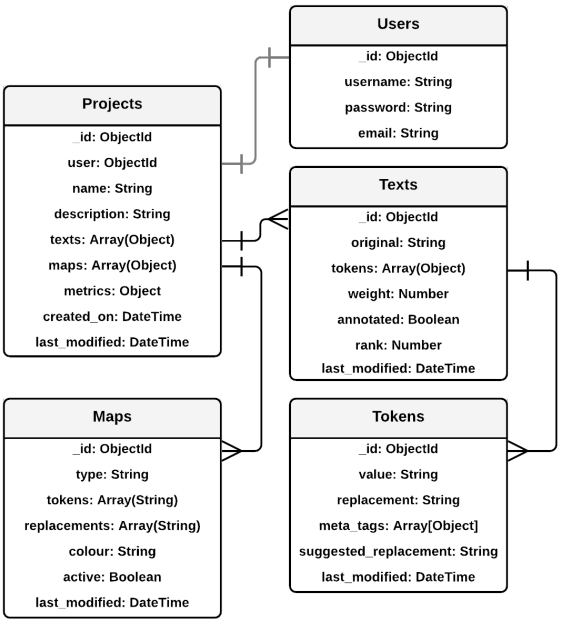

Figure 4: LexiClean's entity relationship diagram.

Here, the `Project` model stores information related to a project including references to `Texts`, `Maps` and `Users`. The `Maps` model captures replacements and meta-tag gazetteers, as well as static assets such as a standard English lexicon. The `Texts` model comprises information pertaining to individual texts such as its original *value*, aggregate tf-idf *weight* and resulting *rank*, whether its been *annotated*, and its constituent *tokens*. `Texts` reference the `Tokens` model that is composed of the tokens original *value*, and accepted or suggested annotations (*replacement*, *meta_tags*, *suggested_replacement*). Lastly, `Users` contains information about users such as their *username*, *password* and *email*.

6 Case Study

Without comparable systems, we demonstrate the efficacy of LexiClean through the annotation of user generated content (UGC) from the specialised-domain of industrial mining (IM) (Sikorska et al., 2016). To date, UGC in industrial domains has received little attention from the NLP community, with state-of-the-art systems relying heavily on hand-craft rules and heuristics for normalisation (Hodkiewicz and Ho, 2016; Gao et al., 2020). More recently, it has also been highlighted that corpora derived from such domains can pose challenges to state-of-the-art NLP systems (Dima et al., 2021).

6.1 Task Setup

We experiment on two corpora (**IM-Pub** and **IM-Priv**) and release one to the public[6]. As LexiClean currently is a single user application, we focus on the performance of a single user annotating under two modes to illustrate the efficacy of LexiClean's features. The two modes are i) from scratch (no automatic labelling using prepopulated replacements or meta-tag gazetteers), and ii) with automatic labelling from prepopulated assets. The same annotator was used for both modes. The annotators native language was English and they had prior familiarity with the domain of industrial mining.

In both modes, OOV token candidates are detected by matching to an English lexicon[7]. Annotation guidelines are borrowed from Baldwin et al. (2015)[8] with extension to support multi-task annotation. For both cases, a set of four meta-tags are used consisting of *domain_specific*, *sensitive*, *unsure* and *noise*. An overview and comparison of the statistics pertaining to both corpora compared to W-NUT15 is shown in Table 1.

	Texts	**Total Tokens**	**OOV Tokens**	
			Count	*Proportion*
IM-Pub	4.5k	21.9k	3.9k	17.8%
IM-Priv	4.5k	21.8k	4.0k	18.3%
W-NUT15	4.9k	73.8k	6.6k	9%

Table 1: Overview and comparison of corpora statistics.

6.2 Case One - Annotation from Scratch

In this case study, annotation of **IM-Pub** is performed, starting from scratch with no automatic

[5]MERN. https://www.mongodb.com/mern-stack

[6]*Industrial Mining Public (IM-Pub).* https://github.com/nlp-tlp/lexiclean/data/im_pub.json

[7]SCOWL (v2020.12.07) English with Australian and British variants (size 60).

[8]*W-NUT15 Guidelines.* http://noisy-text.github.io/2015/norm-shared-task.html

labelling. This corpora consisted of 4.5k texts and 3.9k candidate OOV tokens. The annotator performance shown in Figure 5 highlights the rapidity of OOV token annotation early on in the session owing to features such as cascading corpora wide annotation and the sorting algorithm. The impact of these features is also demonstrated by the user's annotation rate at the start of the session and its increasing nature through to completion. Moreover, a substantial number of normalisations and meta-tags were captured as is evidenced in Table 2.

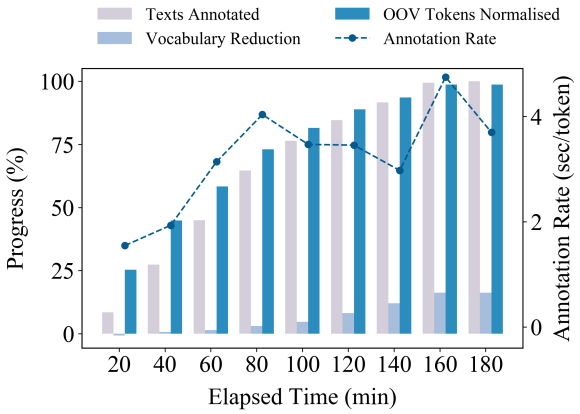

Figure 5: Overview of annotator performance for case one (progress is cumulative).

6.3 Case Two - Annotation from Prepopulated Assets

To evaluate the effectiveness of the automatic labelling feature of LexiClean, annotation of an equivalently sized corpora (**IM-Priv**) to case one was performed. Here, replacements and meta-tag gazetteers generated in case one were exported and used for automatic labelling. It was found that this feature significantly reduced the OOV tokens requiring annotation in IM-Priv by 47% (4,013 to 1,897) as well as reducing the vocabulary size by 3.5%. Comparable with case one, Figure 6 also demonstrated the rapidity of annotation and the ability to apply a significant number of normalisations and associated meta-tags to noisy corpora within a short period (Table 2).

7 Conclusion and Future Work

We have introduced LexiClean, an open-source annotation tool for multi-task lexical normalisation. Stemming from gaps in current token-level annotation tools, we have demonstrated how a dedicated, task-specific tool can enable rapid annotation of

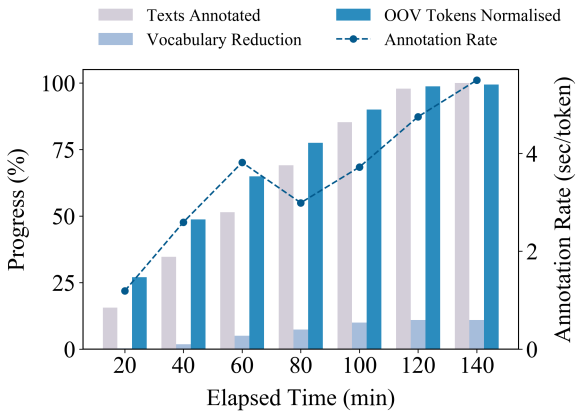

Figure 6: Overview of annotator performance for case two (progress is cumulative).

	Case	One		Two	
	Replacements	706	3967	1025	3168
Meta-tags	*domain_specific*	116	634	245	805
	sensitive	54	382	118	154
	unsure	42	56	111	156
	noise	19	38	29	65

Table 2: Overview of annotation effort for both cases - (*# of unique tokens* | *# of annotated instances*).

large corpora to support both seq2seq and token-classification formulations of the lexical normalisation task. As a result, LexiClean is well positioned to enable future annotation efforts to support the development of the next generation of lexical normalisation algorithms and systems. Future work will focus on converting LexiClean from a single user tool to one that supports multi-user collaborative annotation akin to the current generation of token-level annotation tools.

Acknowledgements

This research is supported by the Australian Research Council through the Centre for Transforming Maintenance through Data Science (grant number IC180100030), funded by the Australian Government. Additionally, Bikaun acknowledges funding from the Mineral Research Institute of Western Australia, and Hodkiewicz acknowledges funding from the BHP Fellowship for Engineering for Remote Operations. Bikaun and Liu acknowledge the support from ARC Discovery Grant DP150102405.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference recent advances in Natural Language Processing RANLP 2013*, pages 198–206.

Alden Dima, Sarah Lukens, Melinda Hodkiewicz, Thurston Sexton, and Michael P. Brundage. 2021. Adapting natural language processing for technical text‡. *Applied AI Letters*, e33.

Yiyang Gao, Caitlin Woods, Wei Liu, Tim French, and Melinda Hodkiewicz. 2020. Pipeline for machine reading of unstructured maintenance work order records. In *Proceedings of the 30th. European Safety and Reliability Conference and 15th. Probabilistic Safety Assessment and Management Conference*. ESRA PSAM.

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.

Melinda Hodkiewicz and Mark Tien-Wei Ho. 2016. Cleaning historical maintenance work order data for reliability analysis. *Journal of Quality in Maintenance Engineering*, 22(2):146–163.

Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2015. Short text understanding through lexical-semantic analysis. In *2015 IEEE 31st. International Conference on Data Engineering*, pages 495–506. IEEE.

Jonathan K Kummerfeld. 2019. Slate: a super-lightweight annotation tool for experts. *arXiv preprint arXiv:1907.08236*.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th. Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 359–367.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Benjamin Muller, Benoît Sagot, and Djamé Seddah. 2019. Enhancing bert for lexical normalization. In *The 5th Workshop on Noisy User-generated Text (W-NUT)*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the 18th. Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

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.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. 2019. Phonetic normalization for machine translation of user generated content. In *Proceedings of the 5th. Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 407–416.

Joanna Sikorska, Melinda Hodkiewicz, Ashwin D'Cruz, Lachlan Astfalck, and Adrian Keating. 2016. A collaborative data library for testing prognostic models. In *PHM Society European Conference*, volume 3.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th. Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Michael Stewart, Wei Liu, and Rachel Cardell-Oliver. 2019a. Redcoat: a collaborative annotation tool for hierarchical entity typing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 193–198.

Michael Stewart, Wei Liu, and Rachel Cardell-Oliver. 2019b. Word-level lexical normalisation using context-dependent embeddings. *arXiv preprint arXiv:1911.06172*.

Michael Stewart, Wei Liu, Rachel Cardell-Oliver, and Rui Wang. 2018. Short-text lexical normalisation on industrial log data. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 113–122. IEEE.

Rob Van der Goot, Barbara Plank, and Malvina Nissim. 2017. To normalize, or not to normalize: The impact of normalization on part-of-speech tagging. *arXiv preprint arXiv:1707.05116*.

Rob van der Goot and Gertjan van Noord. 2017. Monoise: Modeling noise using a modular normalization system. *arXiv preprint arXiv:1710.03476*.

Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2017. Yedda: A lightweight collaborative text span annotation tool. *arXiv preprint arXiv:1711.03759*.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st. Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6.