Kazakh Text Normalization using Machine Translation Approaches

Zhanibek Kozhirbayev[0000-0003-4235-9049] and Zhandos Yessenbayev[0000-0002-6322-3848]

National Laboratory Astana, Nazarbayev University, Nur-Sultan, Kazakhstan
{zhanibek.kozhirbayev, zhyessenbayev}@nu.edu.kz

Abstract. We present herein our work on text normalization applied to user-generated content (UGC) in the Kazakh language collected from Kazakhstani segment of Internet. UGC as a text is notoriously difficult to process due to prompt introduction of neologisms, peculiar spelling, code-switching or transliteration. All of this increases lexical variety, thereby aggravating the most prominent problems of NLP, such as out-of-vocabulary lexica and data sparseness. It has been shown that certain preprocessing, known as lexical normalization or simply normalization, is required for them to work properly.

We applied machine translation techniques to normalize Kazakh texts. For this, a parallel corpus was created with a set of aligned sentences in canonical and non-canonical forms. Using these comments, we created the phrase-based statistical machine translation system as a baseline system. Furthermore, we applied word-based sequence-sequence model to the normalization task. The former method shows 21.67 BLEUs on the test set, whereas later one obtained approximately 30 BLEU score.

Keywords: Text normalization, User-generated content, Sequence-sequence model.

1 Introduction

With the rise of social media, custom text data has reached unprecedented sizes. As part of the project on developing tools and algorithms for processing Kazakh language in the framework of KazNLP project [1, 2], we strive to provide tools for processing real-world data, including user-generated content (UGC). UGC generally refers to any type of content created by Internet users including tweets, comments, dialogues on Internet forums, etc. This type of text is considered difficult to process due to the high level of noise, i.e. it is far from the standards of the literary language. Kazakhstani segment of Internet is not except from noisy UGC and the following cases are the usual suspects in wreaking the “spelling mayhem” [2]:

- spontaneous transliteration, e.g. Kazakh word “біз” can be spelled in three additional ways: “быз”, “биз”, and “biz”;
- use of homoglyphs, e.g. Cyrillic letter “і” (U+0456) can be replaced with Latin homoglyph “i” (U+0069);

- code switching – use of Russian words and expressions in Kazakh text and vice versa;
- word transformations, e.g. “керемееет”,“ крмт” instead of “керемет” (great), or seg-mentation of words, e.g. “к-е-р-е-м-е-т”;
- the use of emoji, e.g. (☺, ☹), and their symbolic counterparts, e.g. [:), : (].

The normalization tool is designed to edit such texts to match the standard language. All these properties of UGC significantly reduce the accuracy of NLP tools, so in practice UGC is often normalized, that is, brought to literary language standards. Consequently, non-canonical text normalization is considered the main preprocessing stage of almost all NLP tasks [3–8].

This paper is organized as follows: Section 2 presents an overview of various techniques for the text normalization task. Data collection and annotation are presented in Section 3. Method description and obtained results of the conducted experiments are described in Section 4. Summary and conclusions of the performed experiments and areas of further research are given in Section 5.

2 Related work

With the rapid growth of content on social media, text normalization has gained increasing attention in the past decade, with a focus on converting noisy non-standard tokens in informal text into standard vocabulary words. Spell checking plays an important role in this process as it can be seen as an initial attempt at text normalization. In [9–11], it was proposed to use a framework with noisy channels to generate a list of corrections for any misspelled word, ranked according to the corresponding posterior probabilities.

The work of [12] refined this structure by computing the likelihood function as a noisy token and its associated tag would be generated by a specific word. However, spell-checking algorithms are in most cases ineffective for this type of data because they do not account for phenomena in informal text. For example, some previous work [13] has instead focused on sporadic typographical errors using edit distance [14] in conjunction with modeling pronunciation.

The work [15] used a noisy channel model based on spell editing distance using the web to generate a large set of automatically generated (noisy) pairs that will be used for training and for spelling suggestions. Even though they use the Web for gathering, they do not focus on informal text, but rather unintentional misspellings. [16] combined the noisy channel model with a rule-based final transformer and obtained acceptable results for French SMS. [17] used weighted finite state machines (FSM) and rewrite rules to normalize French SMS; [18] focused on tweets generated with mobile phones and developed a CRF tagger for deletion-based reduction.

Recent work has also focused on normalizing Twitter messages, which is generally considered a more challenging task. [19] developed classifiers to detect malformed words and generated corrections based on morphophonic similarities. [20, 21] proposed to normalize non-standard tokens without explicitly categorizing them.

The above approaches rely almost heavily on external linguistic resources and manually defined rules. A wide range of NLP tasks shows promising results using neural networks. The encoder-decoder architectures [22, 23] exceeded expectations in machine translation [24], dialogue generation [25], summarization [26], question answering [27]. Hence, it makes sense to wonder if the Seq2Seq models are suitable for the normalization task.

Work of [28] applied the encoder-decoder architecture that uses Recurrent Neural Network (RNN) based on Gated Recurrent Unit (GRU) for Japanese text normalization. They improved the performance of Japanese text normalization by performing a stable training of the encoder-decoder model with a new method for data augmentation. [29] applied a normalization method based on word-character attention-based encoder-decoder model on noisy text in social media. They state that the presented character-based component, which is trained on synthetic adversarial examples, shows a significant result. [30] normalized Swiss German WhatsApp messages using the encoder-decoder model. They argue that the flexibility of the encoder-decoder model provides for using same training data in different ways. Particularly, the modification was made in the part of decoding by introducing different levels of granularity in the language of the target side: characters and words. [31] explored the possibilities of using machine translation techniques to normalize noisy Turkish texts. They trained character-based translation model with synthetic parallel data. The experiments were conducted both on statistical and neural machine translation methods to compare the obtained results.

3 Data collection and annotation

Like most machine learning models, machine translation methods require training data to produce meaningful results. Parallel text corpuses are a structured set of translated texts between two languages. Such parallel corpora are essential for training machine translation algorithms. In our case, the source side is the unprocessed comments, and the target side is the revised comments by annotators. To create a corpus for this task, at the beginning we collected comments from news web pages: nur.kz, tengrinews.kz and zakon.kz. The comments were divided into language groups: Kazakh, Russian and mixed. Perfect comments that do not contain errors have been removed as there is no point in giving correct comments on our machine translation approach. To sort the ideal comments, we used texts from the official news web pages, in which we believe there are no errors. We compare them, if all the words of our comments are in this text, then this comment is considered ideal. Some comments may contain multiple sentences, so we split longer comments into multiple sentences.

The statistics are shown in Table 1.

Table 1. Data set statistics from news portals

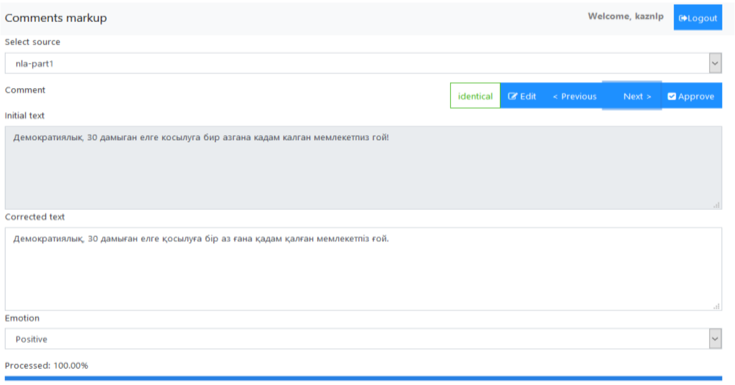
Total		Stripped of perfect comments		After splitting long comments		Ideal comments	
doc	tok	doc	tok	doc	tok	doc	tok
17181	237092	12896	192853	19799	192853	4285	44239

Beside comments from news portals, additional comments were collected from social networks. The Kazakh-speaking audience of commentators is most active on social networks like Facebook and Instagram. The analysis shows the main share of comments in the Kazakh language falls on such Facebook groups as OnlineQazaqstan (367,920 members at the time of analysis), newspaper «Қала мен Дала» (97,685 members at the time of analysis). Based on the above, the Kazakh-speaking segment of the social network Facebook was selected for the source of collecting text data. The statistics of the dataset from social media is presented in Table 2.

Table 2. Social media dataset statistics

Source	Number of posts	Number of comments
OnlineQazaqstan	17	3287
Newspaper «Қала мен Дала»	18	1490
Kaspi.kz	8	1897
Stan.kz	29	3340
Total	72	10 014

After collecting comments, we built a parallel corpus. The source side of the corpus contains comments, and the target side contains revised version of the corresponding comment. A web interface has been built to fix comments and make annotation easier. We had two annotators and one last controller-moderator. Fig. 1 shows a screenshot from the web interface. Here, annotators can select the source of the correction, and can also observe the work done in general. The controller-moderator can correct the work of the annotators and approve.



Select source: nla-part1

Comment

Initial text: Демократиялык, 30 дамыган елге косылуга бир азгана кадам калган мемлекетпиз гой!

Corrected text: Демократиялық, 30 дамыған елге қосылуға бір аз ғана қадам қалған мемлекетпіз ғой.

Emotion: Positive

Processed: 100.00%

Fig. 1. A parallel corpus annotation tool

After the annotation process, the datasets were further processed. Some very long comments are split into several parts, mostly by sentence. Comments in Russian were removed. After these preprocessing, 27005 comments remained. We used 90% of these comments for training and the rest for testing. Statistics are shown in Table 3.

Table 3. Final data statistics

Parallel comments	Train set	Test set
27005	24 305	2700

4 Method description and results

In this project, we explored the potential of using machine translation methods to normalize non-canonical texts in Kazakh. Therefore, we conducted both statistical machine translation (SMT) and neural machine translation (NMT) approaches in order to compare the results.

The SMT method was chosen as a baseline experiment. A pretty standard set of tools was used in this pipeline. The plan was to build scalable NLP tools in Python, so we built a phrase-based statistical machine translation system, since among the various methods, phrase-based methods have shown high performance. We used n-gram language models, in particular 3-gram models. The decoding process was implemented using the beam search stack decoding algorithm.

Inspired by advances in NMT, we applied end-to-end neural network models, in particular sequence-sequence (Seq2Seq) models to the secondary normalization task. Seq2Seq models have ability to convert sequences from one domain (e.g. sentences in non-canonical form) to sequences in another domain (e.g. the same sentences in canonical form). Its feature to capture any useful contextual information in a sentence can be used in text normalization task. This eliminates the need for language-specific tools except the sufficient training data.

We built our Seq2Seq model using the Keras library [32]. Firstly, the combination of the train and test datasets was used to define the maximum length and vocabulary of the problem. We map words to integers, as needed for modeling. Separate tokenizer was used for the source sequences and the target sequences. Each input and output sequence must be encoded to integers and padded to the maximum phrase length, since a word embedding was used for input sequences and one-hot encoding for output sequences.

We use an encoder-decoder Long Short Term Memory (LSTM) networks model on this problem. In this architecture with 2-layer LSTM encoders and decoders, the input sequence is encoded by a front-end model called the encoder then decoded word by word by a backend model called the decoder. The model is trained using the efficient Adam approach to stochastic gradient descent and minimizes the categorical loss function because we have framed the prediction problem as multi-class classification.

To assess the quality of translation, we used a widely used measurement – BLEU (Bilingual Evaluation Understudy) [33]. The main idea behind this metric is to determine the n-gram match between the translated candidate and the link. After translation, we compared our translated test set with an original test set. This result can be viewed in Table 4.

Table 4. Final results

Model	BLEU score
SMT	21.67
NMT	29.74

5 Conclusion

In this work, we used machine translation approaches to normalize comments. To create machine translation systems, we first collected comments from news portals and social networks. We then corrected these comments with annotators. A total of 27005 comments were collected and corrected. The original raw comments are treated as the source side, and the revised comments are treated as the target side in our parallel corpus. Using these comments, we have created the phrase-based statistical machine translation system as a baseline system. Furthermore, we applied word-based sequence-sequence models to the secondary normalization task in order to compare statistical and neural network approaches. The statistical method shows 21.67 BLEUs on the test set, whereas sequence-sequence model obtained approximately 30 BLEU score. The later technique improves the performance of the normalization task significantly. In average, the both results can be viewed as an average performance. The reason for this phenomenon may be related to sparse datasets. To solve this problem, in the future we are going to add more comments to our parallel dataset. Moreover, we will conduct experiments with sequence to sequence models with attention mechanism as well as character-based models.

6 Acknowledgement

This work has been funded by the Ministry of Education and Science of the Republic of Kazakhstan under the research grants No. AP05134272 and No. AP08053085.

References

1. Yessenbayev, Z., Kozhirbayev, Z., Makazhanov, A.: KazNLP: A Pipeline for Automated Processing of Texts Written in Kazakh Language. In: Karpov A., Potapova R. (Eds.): SPECOM 2020, LNAI, 12335, 657–666. Springer Nature Switzerland AG (2020).
2. Makazhanov, A., Yessenbayev, Z., Kozhirbayev, Z.: KazNLP: NLP Tools for Kazakh Language. https://github.com/nlacslab/kaznlp, last accessed 2020/10/07.
3. Kozhirbayev, Z., Yessenbayev, Z., Makazhanov, A.: Document and word-level language identification for noisy user generated text. In: 12th IEEE International Conference on Application of Information and Communication Technologies (AICT2014), 124–127. Almaty (2018).

4. Makazhanov, A., Myrzakhmetov, B., Kozhirbayev, Z.: On Various Approaches to Machine Translation from Russian to Kazakh. In: 5th International Conference on Turkic Languages Processing (TurkLang 2017), 195–209. Kazan (2017).
5. Zulkhazhav, A., Kozhirbayev, Z., Yessenbayev, Z., Sharipbay, A.: Kazakh text summarization using fuzzy logic. Computacion y Sistemas, 23(3), 851–859 (2019).
6. Myrzakhmetov, B., Kozhirbayev, Z.: Extended language modeling experiments for Kazakh. CEUR Workshop Proceedings, 2303, 42–52 (2018).
7. Kozhirbayev, Z., Yessenbayev, Z., Karabalayeva, M.: Kazakh and Russian languages identification using long short-term memory recurrent neural networks. In: 11th IEEE International Conference on Application of Information and Communication Technologies, 342–347. Moscow (2017).
8. Kozhirbayev, Z., Erol, B. A., Sharipbay, A., Jamshidi, M.: Speaker recognition for robotic control via an IoT device. In: World Automation Congress, 259–264. Stevenson, Washington (2018).
9. Church, K.W., Gale, W.A.: Probability scoring for spelling correction. Statistics and Computing, 1(2), 93–103 (1991).
10. Mays, E., Damerau, F. J., Mercer, R. L.: Context based spelling correction. Information Processing & Management, 27(5), 517–522 (1991).
11. Brill, E., Moore, R. C.: An improved error model for noisy channel spelling correction. In: Proceedings of the 38th annual meeting of the association for computational linguistics, 286–293. Hong Kong (2000).
12. Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., Richards, C.: Normalization of non-standard words. Computer speech & language, 15(3), 287–333 (2001).
13. Toutanova, K., Moore, R. C.: Pronunciation modeling for improved spelling correction. In: 40th Annual Meeting of the Association for Computational Linguistics (ACL), 144–151. Philadelphia (2002).
14. Kukich, K.: Techniques for automatically correcting words in text. Acm Computing Surveys (CSUR), 24(4), 377–439 (1992).
15. Whitelaw, C., Hutchinson, B., Chung, G., Ellis, G.: Using the web for language independent spellchecking and autocorrection. In: 2009 Conference on Empirical Methods in Natural Language Processing, 890–899. Singapore (2009).
16. Beaufort, R., Roekhaut, S., Cougnon, L.A., Fairon, C.: A hybrid rule/model-based finite-state framework for normalizing SMS messages. In: 48th Annual Meeting of the Association for Computational Linguistics, 770–779. Uppsala (2010).
17. Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S., Basu, A.: Investigation and modeling of the structure of texting language International Journal of Document Analysis and Recognition (IJDAR), 10(3–4), 157–174 (2007).
18. Pennell, D., Liu, Y.: Normalization of text messages for text-to-speech. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 4842–4845. Dallas (2010).
19. Han, B., Baldwin, T.: Lexical normalisation of short text messages: Makn sens a #twitter. In: 49th annual meeting of the association for computational linguistics: Human language technologies, 368–378. Portland, Oregon (2011).
20. Liu, F., Weng, F., Wang, B., Liu, Y.: Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision. In: 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 71–76. Portland, Oregon (2011).

21. Gouws, S., Metzler, D., Cai, C., Hovy, E.: Contextual bearing on linguistic variation in social media. In: Proceedings of the workshop on language in social media (LSM), 20–29. Portland, Oregon (2011).
22. Sutskever, I., Vinyals, O., Le, Q.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, 3104–3112. Montreal (2014).
23. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078 (2014).
24. Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144. (2016).
25. Vinyals, O., Le, Q.: A neural conversational model. arXiv:1506.05869 (2015).
26. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv:1602.06023 (2016).
27. Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X.: Neural generative question answering. arXiv:1512.01337 (2015).
28. Ikeda, T., Shindo, H., Matsumoto, Y.: Japanese text normalization with encoder-decoder model. In: Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT), 129–137. Osaka (2016).
29. Lourentzou, I., Manghnani, K., Zhai, C.: Adapting sequence to sequence models for text normalization in social media. In: International AAAI Conference on Web and Social Media, 335–345. Munich (2019).
30. Lusetti, M., Ruzsics, T., Göhring, A., Samardžić, T., Stark, E.: Encoder-decoder methods for text normalization. In: Fifth Workshop on NLP for Similar Languages, Varieties and Dialects. Santa Fe, New Mexico (2018).
31. Çolakoğlu, T., Sulubacak, U., Tantuğ, A. C.: Normalizing Non-canonical Turkish Texts Using Machine Translation Approaches. In: 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 267–272. Florence (2019).
32. Chollet, F.: Deep learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG (2018).
33. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: 40th annual meeting of the Association for Computational Linguistics, 311–318. Philadelphia (2002).