Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Neural Text Normalization in Speech-to-Text Systems with Rich Features

Oanh Thi Tran & Viet The Bui

To cite this article: Oanh Thi Tran & Viet The Bui (2021) Neural Text Normalization in Speech-to-Text Systems with Rich Features, Applied Artificial Intelligence, 35:3, 193-205, DOI: 10.1080/08839514.2020.1842108

To link to this article: https://doi.org/10.1080/08839514.2020.1842108

Published online: 11 Jan 2021.

Submit your article to this journal

Article views: 1257

View related articles

View Crossmark data

Citing articles: 6 View citing articles

APPLIED ARTIFICIAL INTELLIGENCE
2021, VOL. 35, NO. 3, 193–205
https://doi.org/10.1080/08839514.2020.1842108

Taylor & Francis
Taylor & Francis Group

Neural Text Normalization in Speech-to-Text Systems with Rich Features

Oanh Thi Tran[a] and Viet The Bui[b]

[a]International School, Vietnam National University, Hanoi, Vietnam; [b]FPT Technology Research Institute, FPT University, Hanoi, Vietnam

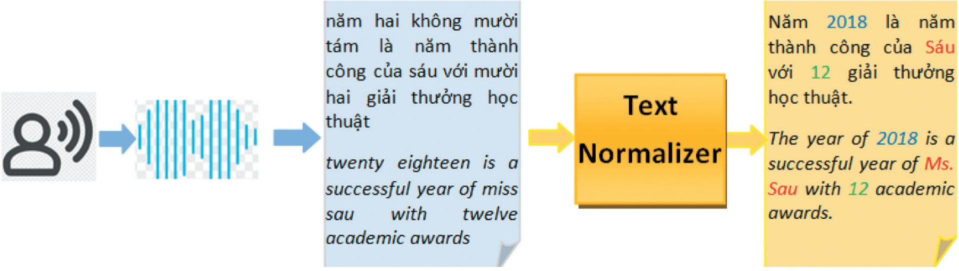
ABSTRACT
This paper presents the task of normalizing Vietnamese transcribed texts in Speech-to-Text (STT) systems. The main purpose is to develop a text normalizer that automatically converts proper nouns and other context-specific formatting of the transcription such as dates, time, and numbers into their appropriate expressions. To this end, we propose a solution that exploits deep neural networks with rich features followed by manually designed rules to recognize and then convert these text sequences. We also introduce a new corpus of 13 K spoken sentences to facilitate the process of the text normalization. The experimental results on this corpus are quite promising. The proposed method yields 90.67% in the F1 score in recognizing sequences of texts that need converting. We hope that this initial work will inspire other follow-up research on this important but unexplored problem.

Introduction

As the name would indicate, Speech-to-Text is a system that gets speech input and instantly generates texts as it is recognized from streaming audio or as the user is speaking. This type of automatic speech recognition systems generally produces un-normalized text (as indicated in Figure 1) which is difficult to read for humans and degrades the performance of many downstream machine processing tasks. Restoring the norm-texts greatly improves the readability of transcripts and increases the effectiveness of subsequent processing, like machine translation, summarization, question answering, sentiment analysis, syntactic parsing, and information extraction, etc. Normalizing transcribed texts, therefore, plays an important role in STT systems. It usually consists of two main tasks:

(1) Punctuator detection which mainly focuses on periods (sentence boundaries).

CONTACT Oanh Thi Tran ✉ oanhtt@isvnu.vn



năm hai không mười tám là năm thành công của sáu với mười hai giải thưởng học thuật

twenty eighteen is a successful year of miss sau with twelve academic awards

Năm 2018 là năm thành công của Sáu với 12 giải thưởng học thuật.

The year of 2018 is a successful year of Ms. Sau with 12 academic awards.

Figure 1. An example of normalizing texts: one year (2018), one person name (Ms. Sáu), and a number (12) are recognized and converted into their written formats.

(2) Automatically recognize and convert the spoken form of texts into their written expressions adhering to a single canonical rule.

In this study, we assume that the former task is solved by using a simple technique based on the information of long silence between speeches, to identify sentence boundary. We, hence, concentrate on the latter task which aims to automatically transcribe proper nouns and typical context-specific formatting. Three main types of proper nouns which are person names, organization names, and location names; and three typical context-specific formattings such as dates, time, and numbers are considered in this research. Such proper nouns and formatting are called entities in this paper.

General speaking, detecting such entities requires quite a bit of linguistic sophistication and native speaker intuition (Schutze (1997)). In the running example of Figure 1, the first occurrence of ‘year’ is ambiguous with the digit ‘five,’ or the person name ‘Sáu’ is ambiguous with the number ‘six’ if their surrounding contexts are not fully considered. Hence, automatically disambiguating these cases is a challenging task because of different ambiguity issues existing in both un-normal texts and other forms of texts.

To our knowledge, it seems that there has no related public research on text normalization in spoken forms of texts so far. Hence, this paper is the first work that formulates the task and provides a preliminary solution to solve it. To this end, the proposed solution is a hybrid architecture that exploits deep neural networks with rich features followed by rule-based processing. Specifically, LSTM (Hochreiter and Schmidhuber (1997); Lample et al. (2016)) and CNN (LeCun et al. (1989)) models integrated with rich manually built features are used to automatically detect these entities. Then, some necessary rules are built to convert these detected entities into their appropriate written expressions. A new corpus consisting of 13 K spoken sentences is also annotated to allow deep learning solutions to be deployed. In conclusion, this paper makes the following contributions:

- Presents the new task of normalizing texts in Vietnamese STT systems.
- Provides a preliminary solution based on neural networks with rich features followed by rules.
- Introduce a new corpus to conduct experiments and facilitate the process of normalizing texts in STT systems.

The rest of this paper is organized as follows. The next section discusses related work. Then, we formally define the problem, and propose a solution to solve it. After that, we introduce our new manually built corpus in a general domain and shows some statistics. Experimental setups, experimental results, and some discussions are also reported. Finally, we conclude the paper and point out some future lines of work.

Related Work

Text normalization is an important stage in processing non-canonical language from natural sources such as social texts, speech, short messages, etc. This is a new research field and most of its papers published are done for popular languages such as English, Japanese, Chinese, etc. All of these text normalization systems usually focus on social texts (Eryigit and Torunoglu-selamet (2017); Ikeda, Shindo, and Matsumoto (2016); Hassan and Menezes (2013)), short messages (Aw et al. (2006)), text-to-speech systems (Yolchuyeva, Gyires-Toth, and Nemeth (2018)), etc. For example, Eryigit and Torunoglu-selamet (2017) present the first work on the social media text normalization of an MRL and introduces the first text normalization system for Turkish. Ikeda, Shindo, and Matsumoto (2016) present a Japanese text normalization using Encoder-Decoder model. Aw et al. (2006) propose a phrase-based statistical model for normalizing SMS texts. For text normalization systems involving speech and language technologies, there have been several works to convert texts from written expressions into their appropriate 'spoken' forms. For example, Yolchuyeva, Gyires-Toth, and Nemeth (2018) introduce a novel CNNs based text normalizer and verify its effectiveness on the dataset of a text normalization challenge on Kaggle.[1]

To our knowledge, there is no public research focusing on the spoken forms of texts in STT systems, especially in Vietnamese. Spoken forms of texts behave quite differently from normal written texts and have some very special phenomena. They are much longer and highly ambiguous (as can be seen in the above examples). To normalize the spoken texts, a straightforward approach is to use predefined rules because they seem to follow some underlying syntactical patterns. These rules can be designed by observing the output of STT systems. However, the approach still poses some disadvantages such as difficult to construct highly accurate rules, time-consuming, need domain-expert skills, difficult to maintain and extend rules, and not really effective.

This is due to the fact that the rule-based approach usually could not deal well with ambiguity problems. To a large extent, it is necessary to consider semantic information of texts and their surrounding contexts.

Rather than using rules, this paper proposes a machine learning-based architecture to solve the task. This approach exploits deep neural networks with rich features followed by some language-specific heuristic rules to recognize and convert text sequences need normalizing into their right formats.

A Solution to Normalize Texts in STT Systems

In this section, we first formally state the problem and then propose a solution to address it.

Problem Statement

The problem can be stated as follows: Given a sequence of syllables which are outputs of an STT system $S = \{s_1, s_2, \ldots, s_n\}$, s_i is the i^{th} syllable (assuming that the output was sentence-segmented), it is required to transcribe S into clean verbatim text formats as follows:

- Capitalize the first letter of the first syllable s_1 in S.
- Capitalize the first letter of each syllable s_k in any proper noun. For our analysis, we consider three common types of proper nouns which are person names, organization names, and location names.
- Capitalize all letters in a syllable s_k if s_k is a course identifier (e.g. VN247) or an abbreviation of organization names (e.g. WHO, FPT, VNPT, etc.).
- Write out numbers zero through ten unless they are part of some cases such as sports records (2–0), time, binary, date, etc. Numbers above 10 represent numerical digits. For numbers above 999.999, substitute million, billion, etc., for the zeros. For dates/time, we use the Vietnamese formats of *dd/MM/yyyy* and *hh:mm:ss*.
- Replace uoms (unit-of-measurements) with their symbols such as Hz, %, USD, etc.

A Proposed Solution to Normalize Texts

The overall architecture is presented in Figure 2 with three main steps as follows:

(1) **Pre-processing**: The texts are pre-processed to capture phone numbers, URLs, e-mail address, … if they follow their formal syntactical patterns.
(2) **Recognizing Proper Nouns and other text formatting**: This is the most critical and difficult step. To automatically recognize if segments

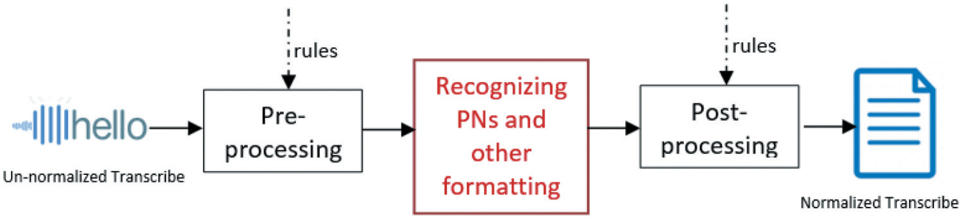


Figure 2. An overall architecture of the text normalization in STT system.

of texts are currently considered as valid entities, it is necessary to know its surrounding contexts. In this work, instead of using heuristic rules, we exploit machine learning techniques by modeling the task as a sequence labeling problem. A fast and effective strategy to label each word is to use its own features to predict labels independently. The best solution is to make the optimal label for a given element dependent on the choices of nearby elements. To this end, we use CRFs (Lafferty, McCallum, and Perera (2001)) which are widely applied, and yield state-of-the-art results in many NLP problems (Bach, Linh, and Phuong (2018); Tran and Luong (2018)).

To build a strong model, CRFs need a good feature set. These features will be automatically learned via neural models and then enriched with manual-built features m. Figure 3 shows the architecture of applying neural architectures to automatically extract useful features for the model. We first use convolutional neural networks by LeCun et al. (1989); Zhu et al. (2017) to encode character-level information of a t^{th} word into its character-level representation l_t. l_t was initialized randomly and trained with the whole network of CNNs. We then combine l_t with word-level representations w_t and manually built features m_t feed x_t = concat (l_t, w_t, m_t) into bi-LSTM networks (Lample et al. (2016); Mikolov et al. (2011)) to model context information of each word. Formally, the formulas to update an LSTM unit at time t are:

$$i_t = \sigma(W_i h_{t-1} + U_i X_t + b_i) \tag{1}$$

$$f_t = \sigma\left(W_f h_{t-1} + U_f X_t + b_f\right) \tag{2}$$

$$\tilde{c}_t = tanh(W_c h_{t-1} + U_c X_t + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

$$o_t = \sigma(W_0 h_{t-1} + U_0 X_t + b_0) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

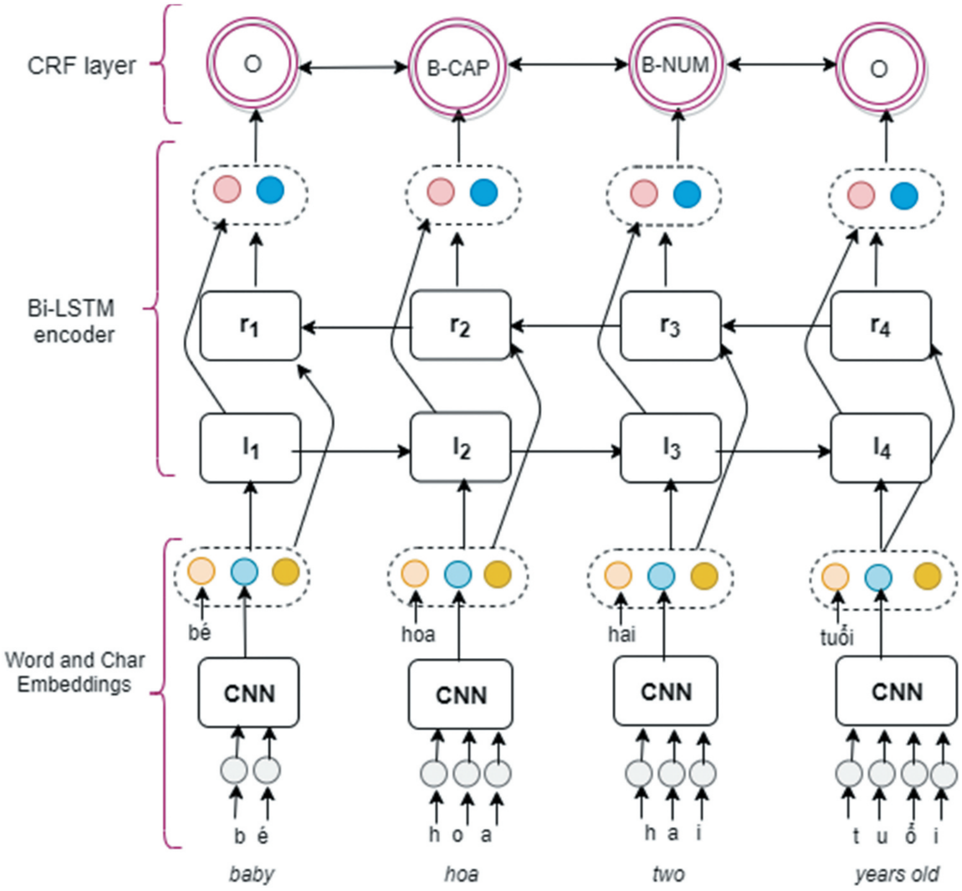


Figure 3. An architecture to detect entities in STT output using biLSTMs.

where σ is the element-wise sigmoid function and $\odot$ is the element-wise product. x_t is the input vector (concatenation of character, word, and manual rich features embeddings) at time t. h_t is the hidden state vector which stores all the useful information up to time t. U_i, U_f, U_c, and U_o denote the weight matrices of different gates for input x_t, and W_i, W_f, W_c, and W_o are the weight matrices for hidden state h_t. b_i, b_f, b_c, and b_o denote the bias vectors. The idea of using bi-LSTMs is to present each sequence forwards and backward to capture past and future information, respectively. These two hidden states l_t and r_t are then concatenated to form the final output. Finally, a CRF is used to take into account neighboring tags, yielding the final context predictions for every word of the input texts.

(3) **Post-processing**: After detected, this step applies some rules to convert these entities into their written expressions as follows:

(a) If a syllable belongs to any recognized proper nouns, just capitalize its first letter. We do a further step to check if this syllable appears in a pre-defined list of course identifiers; we just capitalize every single character of it.

(b) If a sequence of syllables is determined as a date, time, or a number, we built corresponding rules to convert it into its written form. To design these rules, we ask the help of two linguistic experts to write down all possible reading methods/styles in different regions of Vietnam. Then, rules are gradually composed to catch up almost these possible methods.

(c) For uom, a list is manually built to help converting them into their correct symbols.

Experiments

Corpus Building

This section introduces the steps we took to annotate the corpus. We first describe the annotation process, and then show some statistical figures on this corpus. The annotation process is illustrated in Figure 3. It includes the following four main steps:

(1) **Collecting raw data**: Texts from different sources, mainly from online newspapers, were collected.

(2) **Pre-processing**: Like other standard practices, the texts were split into sentences and the main punctuation was removed. During normalization, all words were converted to lowercases, and words with a dash or a colon were separated, keeping the dash and colon as words. Numbers/dates/time/phone numbers/abbreviations/course numbers, etc., were transformed to their full, spoken forms. Then, we randomly selected 13 K sentences and asked two annotators to manually label them with the required information.

(3) **Label designing**: We designed a corresponding set of labels that facilitates the goal of normalizing the output of STT systems. We developed guidelines containing these labels. The guidelines provided detailed examples of the annotations, as well as specific information for each label that helped the annotators easier to annotate and solve conflicts in ambiguous cases. Three labels were chosen are Proper_Noun, Dates/Time, and Numbers.

(4) **Data tagging**: To speed up the labeling step, a tool is built to automatically transform written texts into spoken forms of texts by using some rules such as:

- Phone-similar digit strings are transformed into spoken forms of each discrete digit.

- Some abbreviations are converted into their full forms by using a predefined list of Vietnamese abbreviations. This list is composed of scanning the newspapers to find out abbreviated words (they usually are not valid Vietnamese words). Then, a person is required to manually check and finalize the list.
- Long and short dates/time: We varied different reading methods of these dates/time. Popular reading methods are randomly chosen with higher probabilities.
- Numbers: similar to dates/time, we also diversify different reading methods for each number.

Then, we hire two annotators to manually check and correct wrong labels and unnatural reading of a given text using the predefined set of labels designed in the previous step. To measure the inter-annotator agreement, we used Cohen's kappa coefficient (Cohen (1960)). Some statistics about the corpus are given in Table 1. Cohen's kappa coefficient of our corpus was 0.91, which usually is interpreted as almost perfect agreement.

Experimental Setups

To create word embeddings, we collected the raw data from Vietnamese newspapers (~9GB texts) to train the word vector model using Glove[2] (Pennington, Socher, and Manning (2014)). We fixed the number of word embedding dimensions at 50, the number of character embedding dimensions at 25. We also defined our own features and then mapped them into a vector of 10 dimensions.

For each experiment type, we conducted fivefold cross-validation tests. The hyper-parameters were chosen via a search on the development set. We randomly select 10% of the training data as the development set. The system performance is evaluated using precision, recall, and the F_1 score as in many sequence labeling problems (Yadav and Bethard (2018)) as follows:

$$F_1 = \frac{2 * pre * rec}{pre + rec} \tag{7}$$

$$pre = \frac{TP}{TP + FP} \tag{8}$$

Table 1. Some statistics about the corpus.

No.	Entities	#of Samples
1	Proper Nouns	19.744
2	Dates/Time	1.372
3	Number	7.006
4	#of Sentences	13.000

$$rec = \frac{TP}{TP + FN} \tag{9}$$

where TP (True Positive) is the number of entities that are correctly identified. FP (False Positive) is the number of text sequences that are mistakenly identified as valid entities. FN (False Negative) is the number of entities that are not identified.

Experimental Results

In this section, we presented two types of experiments. The first one is to evaluate the effectiveness of the proposed method in detecting entities that need converting. This type includes three experiments to evaluate the baseline, the proposed method with/without using rich features. The second type is to integrate this text normalizer into a real STT system to measure its final performance on real output texts.

Experimental Results of the Baseline Using Rules to Detect Entities

Based on some dictionaries about Vietnamese person names, location names, organization names, we designed some rules to captures these proper nouns automatically. For dates/time and numbers, we try to capture some popular reading ways of human beings among different regions of Vietnam. These rules are implemented by using the module *re* of Python language. Table 2 shows its experimental results.

The baseline had a higher precision than recall in general due to the fact that if a match is found it is probably correct. It got a precision of 82.64%, a recall of 84.07% and an F1 score of 83.36% averaged on five folds.

Experimental Results of the Proposed Model without Using Rich Features

Table 3 illustrates the experimental results of the proposed model. As can be seen that the proposed method got much higher results. The recall, precision, and F1 scores are significantly increased on all five folds. Overall, it can greatly improve the efficiency of recognizing these entities. Specifically, compared to the baseline, this method remarkably boosted the F1 metric by 4.11%, precision by 6.9%, and recall by 1.43%.

Table 2. Experiment results using the rule-based baseline to detect entities.

No.	Pre	Rec	F_1
Fold 1	82.66	84.21	83.43
Fold 2	82.69	84.15	83.41
Fold 3	82.47	83.83	83.14
Fold 4	82.84	84.18	83.50
Fold 5	82.55	84.06	83.30
Average	82.64	84.07	83.36

Table 3. Experiment results using the neural architectures without rich features.

No.	Pre	Rec	F_1
Fold 1	89.62	85.18	87.35
Fold 2	89.67	85.70	87.64
Fold 3	89.79	84.86	87.25
Fold 4	89.06	85.23	87.10
Fold 5	89.51	86.55	88.00
Average	89.53	85.50	87.47

Table 4. Experiment results using the neural network with rich features.

No.	Pre	Rec	F_1
Fold 1	93.38	88.09	90.66
Fold 2	93.73	88.42	91.00
Fold 3	93.31	87.98	90.57
Fold 4	93.63	87.89	90.67
Fold 5	93.94	87.23	90.46
Average	93.60	87.92	90.67

Experimental Results of the Proposed Model Integrated with Rich Features

Table 4 shows the experimental results using rich features. The results suggested that integrating rich features into neural models boosts the performance of the final system in detecting entities. In comparison to not using rich features, it increased the precision by 4.07%, the recall by 2.42%, and the F1 score by 3.2%, respectively.

Figures 4 and 5 shows experimental results of the F1 scores on each label. As can be seen that numbers are easiest to detect, followed by dates/time. Proper nouns are the most difficult to recognize because they are highly ambiguous.

Experimental Results of the Final System on the Output of a Real STT System

We integrated the best entity recognition model using biLSTM with rich features into our STT system to test its performance. The evaluation of the model was performed on 1000 real-world examples. Testers were required to read these randomly selected sentences as inputs and collect the outputs. Then, we also measured precision, recall, and F1 scores based on the numbers of the entities such as proper nouns, dates/time, and numbers. The experimental results show that the text normalizer yields 75.78% in precision, 82.2% in recall, and 78.86% in the F1 score.

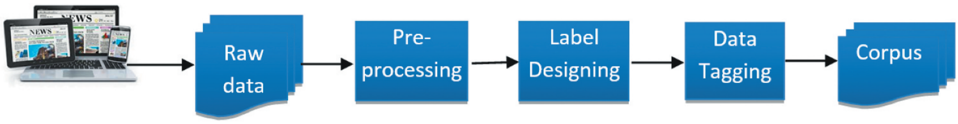


Figure 4. The annotation process in building the new corpus.

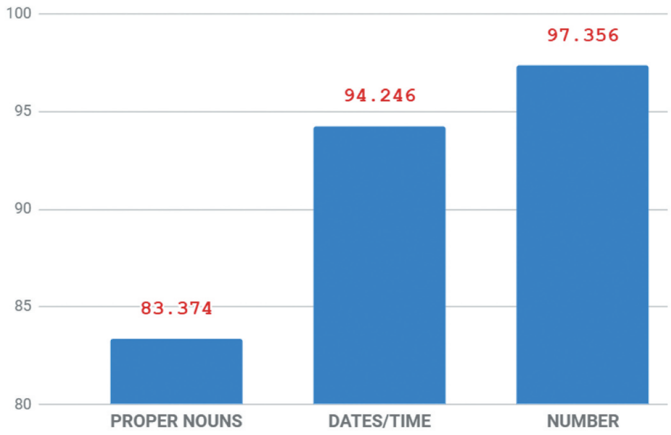
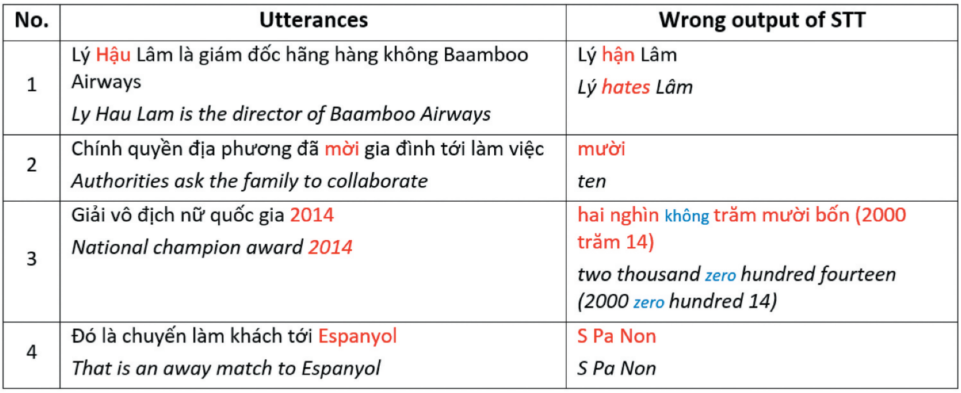


Figure 5. Performance on each label in the F1 scores (in %).



No.	Utterances	Wrong output of STT
1	Lý Hậu Lâm là giám đốc hãng hàng không Baamboo Airways *Ly Hau Lam is the director of Baamboo Airways*	Lý hận Lâm *Lý hates Lâm*
2	Chính quyền địa phương đã mời gia đình tới làm việc *Authorities ask the family to collaborate*	mười *ten*
3	Giải vô địch nữ quốc gia 2014 *National champion award 2014*	hai nghìn không trăm mười bốn (2000 trăm 14) *two thousand zero hundred fourteen (2000 zero hundred 14)*
4	Đó là chuyến làm khách tới Espanyol *That is an away match to Espanyol*	S Pa Non *S Pa Non*

Figure 6. Some examples of wrong outputs of STT integrated with the text normalizer. (It highlights wrong expressions in red).

These results are lower than previous experiments on ideal spoken forms of texts. The main reason is that the real output of the STT produces some wrong words which make our text normalizer could not detect the sequence of texts that need converting. Some examples are shown in Figure 6. The first case shows the wrong output of the middle name of a person. In the second case, the verb ‘ask’ was wrongly recognized as a number ‘ten.’ The third case shows an example of elision where a syllable ‘zero’ is dropped in the STT system. Observing the data, we saw several English words not correctly produced by STT system. This problem also causes a decrease in our text normalizer‘s performance as shown in the fourth case. There are also other types of errors which the entity recognizer could not detect out the right proper nouns, dates, and time.

Conclusion

We presented the first text normalization system for Vietnamese STT using deep neural architectures with rich features followed by manually designed rules. The neural architecture uses CNNs to encode character contexts of a word. Then, we concatenate them with pre-trained word embeddings and rich features to feed into a bi-LSTM encoder. A CRF is then applied on the top to predict label for each word. To conduct experiments, a newly built corpus is also presented for Vietnamese to facilitate the process of normalizing the output of STTs. This new dataset can serve as a benchmark for this task in Vietnamese. Experimental results on this corpus were promising. We achieved 90.67% in the F1 score in recognizing segments of texts that need normalizing, and 78.86% in the F1 score when tested on the real output of an STT system.

Through extensive experiments on this dataset, we acknowledge several insights such as using machine learning techniques is more robust and effective than rules in detecting entities in STT output texts, and some types of entities (e.g. numbers) are easier to detect than others (e.g. proper nouns). We hope that this initial study will inspire other follow-up research on this important but unexplored problem.

Acknowledgement

This research is funded by International School, Vietnam National University, Hanoi (VNU-IS) under project number CS.NNC/2020-05.

Notes

1. https://www.kaggle.com/c/text-normalization-challenge-english-language
2. https://github.com/standfordnlp/GloVe

References

Aw, A., M. Zhang, J. Xiao, and J. Su. 2006. A phrase-based statistical model for SMS text normalization. In *The COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, Sydney, Australia, 33–40.

Bach, N. X., N. D. Linh, and T. M. Phuong. 2018. An empirical study on POS tagging for vietnamese social media text. *Computer Speech & Language* 50:1–15. doi:10.1016/j.csl.2017.12.004.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Journal Educational and Psychological Measurement* 20 (1):37–46. doi:10.1177/001316446002000104.

Eryigit, G., and D. Torunoglu-selamet. 2017. Social media text normalization for Turkish. *Journal of Natural Language Engineering* 23 (6):835–75. doi:10.1017/S1351324917000134.

Hassan, H., and A. Menezes. 2013. Social text normalization using contextual graph random walks. In *51st Annual Meeting of the Association for Computational Linguistics*, Bulgaria, vol. 1, 1577–86.

Hochreiter, S., and J. Schmidhuber. 1997. Long short-term memory. *Journal Neural Computation* 9 (8):1735–80. doi:10.1162/neco.1997.9.8.1735.

Ikeda, T., H. Shindo, and Y. Matsumoto. 2016. Japanese text normalization with encoder-decoder model. In *The COLING 2016 Organizing Committee, Proceedings of the 2nd Workshop on Noisy Usergenerated Text (WNUT*), Osaka, Japan, 129–37.

Lafferty, J. D., A. McCallum, and F. C. N. Perera. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conference on Machine Learning*, 282–89. San Francisco: Morgan Kaufmann Publishers Inc.

Lample, G., M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. 2016. Neural architectures for named entity recognition. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260–70. San Diego: Association for Computational Linguistics.

LeCun, Y., B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. 1989. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2*, 396–404. CA, United States: Morgan-Kaufmann Publisher.

Mikolov, T., S. Kombrink, L. Burget, J. Ernock, and S. Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*), 5528–31. Prague, Czech Republic.

Pennington, J., R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *The 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP*), vol 14, 1532–43. Doha, Qatar: Association for Computational Linguistics publisher.

Schutze, H. 1997. *Ambiguity Resolution in Language Learning: Computational and Cognitive Models*, 176. Stanford, CA : CSLI Publications.

Tran, T. O., and C. T. Luong. 2018. Towards understanding user requests in AI bots. In *15th Pacific Rim International Conference on Artificial Intelligence (PRICAI*), 864–77, Nanjing, China.

Yadav, V., and S. Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *The 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, 2145–58.

Yolchuyeva, S., B. Gyires-Toth, and G. Nemeth. 2018. Text normalization with convolutional neural networks. *International Journal of Speech Technology* 21 (4):589–600. doi:10.1007/s10772-018-9521-x.

Zhu, Q., X. Li, A. Conesa, and C. Pereira. 2017. GRAM-CNN: A deep learning approach with local context for named entity recognition in biomedical text. *Journal Bioinformatics* 34 (9):1547–54. doi:10.1093/bioinformatics/btx815.