A NEURAL ARCHITECTURE TO NORMALIZE TRANSCRIBED TEXTS IN STT SYSTEMS

Oanh Thi Tran[1,2], Viet The Bui[2], and Trung The Tran[2]

[1] International School, Vietnam National University, Cau Giay - Hanoi - Vietnam
[2] FPT Technology Research Institute, Cau Giay - Hanoi - Vietnam
oanhtt@isvnu.vn, vietbt@fpt.com.vn, trungtt3@fpt.com.vn

Abstract. This paper presents the task of normalizing Vietnamese transcribed texts in Speech-to-Text (STT) systems. The main purpose is to automatically convert proper nouns and other context-specific formatting of the transcription such as dates, time and numbers. To this end, we propose a solution which exploits deep neural networks followed by manually-designed rules to convert text sequences from spoken forms into appropriate written expressions. We also introduce a new corpus of 13K spoken sentences to facilitate the process of text normalization. The experimental results on this corpus are quite promising. The proposed method yields 87.47% in the F1 score in recognizing sequences of texts need converting. When testing with the real output of STT systems, this text normalizer yields 78.86% in the F1 score in detecting and then converting these text sequences into written expressions. We hope that this initial work will inspire other follow-up research on this important but unexplored problem.

Keywords: Text normalization · Neural architecture · post-processing STT outputs.

1 Introduction

As the name would indicate, STT is a system which takes speech input and immediately returns texts as its recognized from streaming audio or as the user is speaking. As can be seen that automatic speech recognition systems generally produce 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:

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

– 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 later 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 four typical context-specific formatting such as dates, time, number, and phone numbers are considered in this research. Such proper nouns and formatting are called entities in this paper.

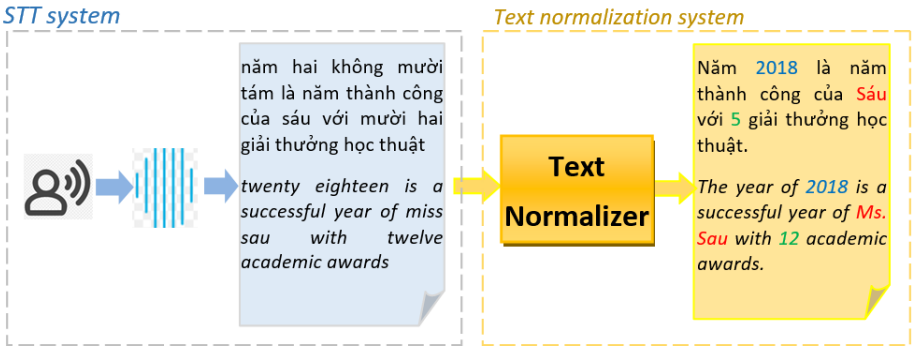

Fig. 1. An example of normalizing texts: one year (2018), one person name (Ms. Sau), and one number (12) are recognized and transcribed into their right written formats..

General speaking, to detect entities mentioned in texts, it requires quite a bit of linguistic sophistication and native speaker intuition [13]. In the above example, 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 not a trivial 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 in spoken forms of texts so far. This is also a new research topic for Vietnamese STT. Hence, this paper formulates and provides a preliminary solution to the task of transcribing proper nouns and other context-specific formatting in Vietnamese STT system. To this end, we propose a hybrid architecture which combines deep neural networks and rule-based approaches. This approach exploits LSTMs[7, 9] and CNNs[10] models to automatically detect these entities, followed by handcrafted rules to automatically convert these detected entities into appropriate written expressions. A new corpus consisting of 13K spoken sentences is also manually built to allow deep learning solutions to be deployed. In conclusion, this paper makes the following contributions:

- Presents a new task of correctly transcribing proper nouns and context-specific formatting in Vietnamese STT systems.
- Provides a preliminary solution based on neural architectures and pre-defined rules.
- Introduce a newly-built corpus to conduct experiments and facilitate the process of normalizing spoken forms of texts into their correct written forms.

The rest of this paper is organized as follows. Section 2 discusses related work. In Section 3, we formally define the problem, and then propose a solution to solve it. Section 4 introduces our new manually-built corpus in a general domain and shows some statistics. Experimental setups, experimental results, and some discussions are reported in Section 5. Finally, we conclude the paper and point out some future lines of work in Section 6.

2 Related Work

Text normalization is an indispensable stage in processing non-canonical language from natural sources, such as speech, social media or short text messages, etc. This is a new research field and its public papers are mostly conducted for popular languages such as English, Turkish, Japanese, etc. All of these text norm systems focus on social media texts [4–6], sms [2], text-to-speech system [16], etc. For example, Eryigit et al., [4] introduce the first work on the social media text normalization of an MRL and presents the first complete social media text normalization system for Turkish. Ikeda et al., [5] present a Japanese Text Normalization with Encoder-Decoder Model. Aw et al., [2] propose a phrase-based statistical model for SMS text normalization. For text normalization systems involving speech and language technologies, there has been several work to convert texts from written expressions into appropriate 'spoken' forms. For example, Yolchuyeva et al., [16] propose a novel CNNs based text normalization method and verify its effectiveness on the dataset of a text normalization challenge on Kaggle[3].

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 into appropriate written expressions, 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, in order to recognize proper

[3] https://www.kaggle.com/c/text-normalization-challenge-english-language

nouns and other context-specific formatting, it is necessary to consider semantic information of texts and their surrounding contexts.

To normalize texts in STT systems, rather than using rules we propose a machine learning-based architecture to solve the task. This approach exploits deep neural networks followed by some language specific heuristics rules to recognize and convert text sequences need normalizing into their right formats.

3 A Hybrid Solution to Normalize Texts in STT Systems

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

3.1 Problem Statement

The problem can be stated as follows: Given a sequence of syllables which are outputs of a STT system $S = \{s_1, s_2, ..., s_n\}$, s_i is the i^{th} syllable (assuming that the output was sentence-segmented), it is required to transcribe them into clean verbatim text formats so that end-users can easily comprehend the output texts. Specifically, a text normalizer should have the ability to:

- Capitalize the first letter of the first syllable s_1 in a sentence.
- 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, KR156, etc.), 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 ten represent with 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, %, $, etc.)

3.2 A hybrid architecture to normalize transcribed texts

To deal with the task, we propose a hybrid solution which exploits deep neural networks followed by rule-based processing. The overall architecture is presented in Figure 2 with three main steps as follows:

1. **Pre-processing**: The transcribed texts are pre-processed to capture phone numbers, URLs, email address, ... if they follow their formal syntactical patterns.

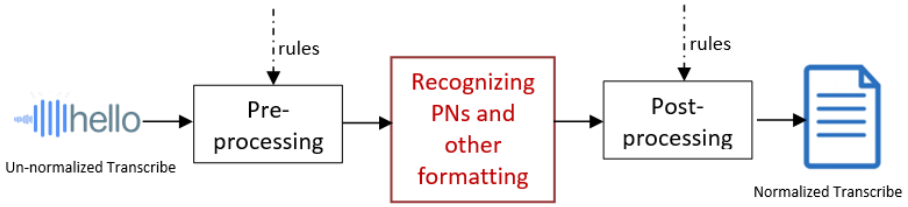

Fig. 2. An overall architecture of the text normalization in STT system.

2. **Recognizing Proper Nouns and other text formatting**: This is the most critical and difficult step. To automatically recognize if segments of texts are currently considered as valid entities, it is necessary to know its surrounding contexts. In this work, instead of using heuristics 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 [8] which are widely applied, and yield state-of-the-art results in many NLP problems [1, 14].
To build the strong model, CRFs need a good feature set. These features will be automatically learnt via neural models. 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 [10] 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 feed $x_t = concat(l_t, w_t)$ into bi-LSTM networks[9] to model context information of each word. Formally, the formulas to update an LSTM unit at time t are:

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

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

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

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

$$o_t = \sigma(W_o h_{t-1} + U_o X_t + b_o) \tag{5}$$

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

where σ is the element-wise sigmoid function and $\odot$ is the element-wise product. x_t is the input vector (concatenation of character and word embeddings) at time t. h_t is the hidden state vector storing all the useful information at (and before) time t. U_i, U_f, U_c, U_o denote the weight matrices of different gates for input x_t, and W_i, W_f, W_c, W_o are the weight matrices for hidden state h_t. b_i, b_f, b_c, b_o denote the bias vectors.

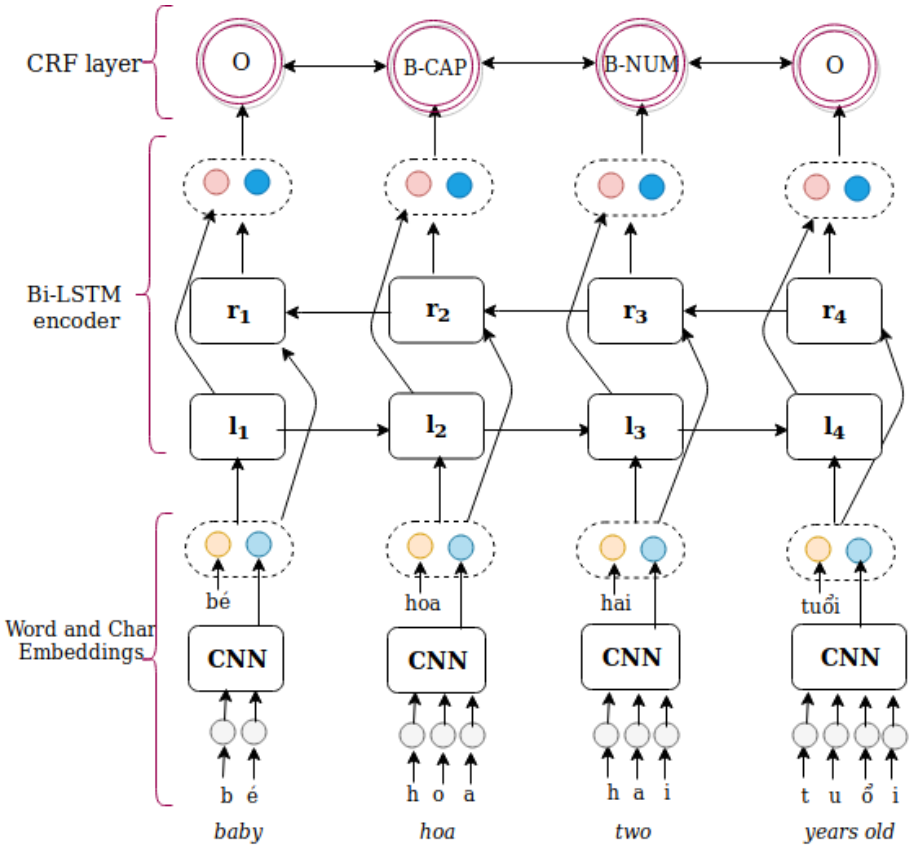

Fig. 3. A neural architecture using biLSTM-CRF to encode features.

The idea of using bi-LSTMs is to present each sequence forwards and backwards to capture past and future information, respectively. These two hidden states are concatenated to form the final output. We then use a CRF to take into account neighboring tags, yielding the final context predictions for every word.

3. **Post-processing**: After detected, this step applies manually-designed rules to convert these entities into their written expressions as follows:
   - If a syllable belongs to any recognized proper nouns (determined in Step 1), 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.
   - If a sequence of syllables is determined as a date, time, or a number, we built corresponding rules to convert 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.
   - For uom, a list is manually built to help converting them into their symbols.

4 Experiments

4.1 Corpus Building

This section introduces the steps we took to annotate the corpus with proper nouns and other formatting to facilitate the process of normalizing transcribed texts in Vietnamese STT systems. We first describe the annotation process, and then show some statistical figures on this corpus.

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

The annotation process is illustrated in Figure 4. 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 13K sentences and asked two annotators to manually label them with required information.

3. **Label Designing**: We designed a corresponding set of labels which facilitates the goal of normalizing the output of STT systems. We developed guidelines containing these labels. The guidelines provided examples of the annotations, as well as specific information for each label that helped the annotators to agree 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 manually-built rules such as:
   - Phone-similar digit strings are transformed into spoken forms of each discrete digit.
   - Some abbreviations: a predefined list of Vietnamese abbreviations is used to convert a given abbreviation into its full form. This list is composed by 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 the Cohens kappa coefficient [3]. Some statistics about the corpus is given in Table 1. The Cohens kappa coefficient of our corpus was 0.91, which usually is interpreted as almost perfect agreement.

Table 1. Some statistics about the corpus.

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

4.2 Experimental Setups

To create word embeddings, we collected the raw data from Vietnamese newspapers (≈ 9GB texts) to train the word vector model using Glove[4] [12]. We

[4] https://github.com/standfordnlp/GloVe

fixed the number of word embedding dimensions at 50, the number of character embedding dimensions at 25.

For each experiment type, we conducted 5-fold 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 [15] as follows:

$$F1 = \frac{2*precision*recall}{precision+recall}$$
$$precision = \frac{TP}{TP+FP}$$
$$recall = \frac{TP}{TP+FN}$$

where TP (True Positive) is the number of entites 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.

We set dropout rate of 0.5, mini-batch size of 50 for CNNs. For bi-LSTMs, we set the number of epochs equals 100, the optimizer as SGD, the batch size as 20, early stopping as $True$ with 4-epoch patience, dropout rate of 0.5.

4.3 Experimental Results

In this section, we presented three types of experiments. Firstly, we established and implemented a baseline using rules to capture proper nouns, dates/time and numbers in spoken forms of texts. The second one is to evaluate the efficiency of using deep neural networks in comparison to the baseline. The last one is to integrate this text normalizer into our STT system to measure its final performance on real outputs of the STT system.

4.4 Experimental results of the baseline using hand-crafted rules

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 5 folds.

4.5 Experimental results of the proposed model

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

Table 2. Experiments results using the rule-based baseline.

No.	Precision	Recall	F1-score
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**

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 3. Experiments results using the neural architectures.

No.	Precision	Recall	F1-score
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**

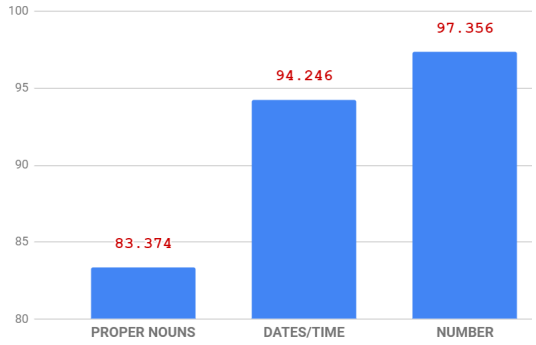

Fig. 5. Performance on each label in the F1 scores (in %).

Figure 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.

4.6 Experimental results of the final system on the output of a real STT system

The evaluation of the model was performed on 1000 real-world examples. Testers were required to read these randomly-selected sentences as inputs for a STT and collect the outputs. Then, we measured precision, recall, and F1 scores based on the numbers of the entities such as proper nouns, dates/time and numbers. The experimental results shows that the text normalizer yields 75.78% in precision, 82.2% in recall, and 78.86% in the F1 score.

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*

Fig. 6. Some examples of wrong outputs of STTs integrated with our text normalizer

These results are lower than previous experiments on ideal spoken forms of texts. The main reason is that the output of the STT produces some wrong words which make our text normalizer could not detect the sequence of texts 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 the decrease of our text normalizer 's performance as shown in the fourth case.

5 Conclusion

We presented the first text normalization system for Vietnamese STT using deep neural architectures combined with manually-deisgned rules. The neural architecture uses CNNs to encode character contexts of a word. Then, we concate them with pre-trained word embeddings 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 87.47% in the F1 score in recognizing segments of texts need normalizing, and 78.86% in the F1 score when tested on a real output of STT systems.

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.

References

1. Bach, N.X, Linh, N.D., Phuong, T.M.: An Empirical Study on POS Tagging for Vietnamese Social Media Text. Computer Speech and Language, **50**, pages 1–15 (2018)
2. Aw, A., Zhang, M., Xiao, J., and Su, J.: A phrase-based statistical model for sms text normalization. In: The COLING/ACL on Main conference poster sessions, pages 33–40. Association for Computational Linguistics (2006)
3. Cohen, J.: A coefficient of agreement for nominal scales. Journal Educational and Psychological Measurement, **20**(1), pages 37–46. doi:10.1177/001316446002000104 (1960)
4. Eryigit, G. and Torunoglu-selamet, D.: Social media text normalization for Turkish. Journal Natural Language Engineering, **23**(6), pages 835–875. doi:10.1017/S1351324917000134 (2017)
5. Ikeda, T., Shindo, H., and Matsumoto, Y.: Japanese text normalization with encoder-decoder model. In: The COLING 2016 Organizing Committee, Proceedings of the 2nd Workshop on Noisy Usergenerated Text (WNUT), pages 129–137 (2016)
6. Hassan, H. and Menezes, A.: Social text normalization using contextual graph random walks. In: 51st Annual Meeting of the Association for Computational Linguistics, volume 1, pages 1577–1586 (2013)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Journal Neural Computation **9**(8), pages 1735–1780 (1997)
8. Lafferty, J.D., McCallum, A., Perera, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: 18th International Conference on Machine Learning, pages 282–289. Morgan Kaufmann Publishers Inc., San Francisco (2001)
9. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270. Association for Computational Linguistics, San Diego (2016)
10. LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D.: Handwritten Digit Recognition with a Back-Propagation Network. In Advances in Neural Information Processing Systems 2, Morgan-Kaufmann Publisher, pages 396–404 (1989)

11. Mikolov, T., Kombrink, S., Burget, L., ernock, J., Khudanpur, S.: Extensions of recurrent neural network language model. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, Prague, Czech Republic, pages 5528–5531 (2011)
12. Pennington, J., Socher, R., and Manning, C.D.: Glove: Global vectors for word representation. In the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), volume 14, Association for Computational Linguistics publisher, pages 1532–1543 (2014)
13. Schutze, H.: Ambiguity Resolution in Language Learning: Computational and Cognitive Models. CSLI Publications, 176 pages (1997)
14. Tran, T.O, Luong, C.T.: Towards Understanding User Requests in AI Bots. In 15th Pacific Rim International Conference on Artificial Intelligence (PRICAI), pages 864–877 , Nanjing, China, (2018)
15. Yadav, V., Bethard, S.: A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In: the 27th International Conference on Computational Linguistics, pages 2145–2158, (2018)
16. Yolchuyeva, S., Gyires-Toth, B., Nemeth, G.: Text normalization with convolutional neural networks. International Journal of Speech Technology **21**(4), May (2018)
17. Zhu, Q., Li, X., Conesa, A., Pereira, C.: GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. Journal Bioinformatics, **34**(9), pages 1547–1554 (2017)