# Spelling Correction as a Foreign Language

Yingbo Zhou*
eBay Inc
San Jose, California
yingbzhou@ebay.com

Utkarsh Porwal
eBay Inc
San Jose, California
uporwal@ebay.com

Roberto Konow
eBay Inc
San Jose, California
rkonow@ebay.com

## ABSTRACT

In this paper, we reformulated the spelling correction problem as a machine translation task under the encoder-decoder framework. This reformulation enabled us to use a single model for solving this problem that is traditionally formulated as learning a language model and an error model. This model employs multi-layer recurrent neural networks as an encoder and a decoder. We demonstrate the effectiveness of this model using an internal dataset, where the training data is automatically obtained from user logs. The model offers competitive performance as compared to the state of the art methods but does not require any feature engineering nor hand tuning between models.

## KEYWORDS

Spell Correction; Encoder Decoder; Noisy Channel

**ACM Reference Format:**
Yingbo Zhou, Utkarsh Porwal, and Roberto Konow. 2019. Spelling Correction as a Foreign Language. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR 2019 eCom).* ACM, New York, NY, USA, 4 pages.

## 1 INTRODUCTION

Having an automatic spelling correction service is crucial for any e-commerce search engine as users often make spelling mistakes while issuing queries. A correct spelling correction not only reduces the user's mental load for the task, but also improves the quality of the search engine as it attempts to predict user's intention. From a probabilistic perspective, let $\tilde{\mathbf{x}}$ be the misspelled text that we observe, spelling correction seeks to uncover the true text $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}|\tilde{\mathbf{x}})$. Traditionally, spelling correction problem has been mostly approached by using the noisy channel model [11]. The model consists of two parts: 1) a language model (or source model, i.e. $P(\mathbf{x})$) that represent the prior probability of the intended correct input text; and 2) an error model (or channel model, i.e. $P(\tilde{\mathbf{x}}|\mathbf{x})$) that represent the process, in which the correct input text got corrupted to an incorrect misspelled text. The final correction is therefore obtained by using the Bayes rule, i.e. $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x})P(\tilde{\mathbf{x}}|\mathbf{x})$. There are several problem with this approach. First, we need two separate models and the error in estimating one model would affect the performance of the final output. Second, it is not easy to model the channel since there is a lot of sources for spelling mistakes, e.g. typing too fast, unintentional key stroke, phonetic ambiguity among others. Lastly, in certain context (e.g. in a search engine) it is not easy to obtain clean training data for language model as the input does not follow what is typical in natural language.

Since the goal is to get text that maximize $P(\mathbf{x}|\tilde{\mathbf{x}})$, can we directly model this conditional distribution instead? In this work, we explore this route, which by passes the need to have multiple models and avoid getting errors from multiple sources. We achieve this by applying the sequence to sequence learning framework using recurrent neural networks [16] and reformulate the spelling correction problem as a neural machine translation problem, where the misspelled input is treated as a foreign language.

## 2 RELATED WORK

Spelling correction is used in wide range of applications other than Web search [4] and e-commerce search such as personal search in email [7] to improve healthcare inquiries [13]. However, noisy channel model or its extensions remain a popular choice for designing large scale spelling correction system. Gao et al. [6] proposed an extension of the noisy channel model where the language model was scaled to Web scale and a distributed infrastructure to facilitate such scaling was proposed. They also proposed a phrase based error model. Similarly, Whitelaw et al. [17] also designed a large scale spelling correction and autocorrection system that did require any manually annotated training data. They also designed their large scale system following the noisy channel model where they extended one of the earliest error models proposed by Brill et al. [3]. Spelling correction problem has been formulated in several different novel ways. Li et al. [12] used Hidden Markov Models to model spelling errors in a unified framework. Likewise, Raaijmakers et al. [14] used deep graphical model for spelling correction. They formulated spelling correction as a document retrieval problem where words are documents and for a misspelled query one has to retrieve the appropriate document. Eger et al. [5] formulated spelling correction problem as a subproblem of the more general string-to-string translation problem. Their work is similar to ours in spirit but differs significantly in implementation detail. We formulate the spelling correction as a machine translation task and to the best of our knowledge no other study has been conducted doing the same.

## 3 BACKGROUND AND PRELIMINARIES

The recurrent neural network (RNN) is a natural extension to feed-forward neural network for modeling sequential data. More formally, let $(x_1, x_2, \ldots, x_T), x_t \in \mathbb{R}^d$ be the input, an RNN update its

[*] This work was done when author worked at eBay

*SIGIR 2019 eCom, July 2019, Paris, France*

internal recurrent hidden states by doing the following computation:

$$h_t = \psi(h_{t-1}, x_t) \tag{1}$$

where $\psi$ is a nonlinear function. Traditionally, in a standard RNN the $\psi$ is implemented as an affine transformation followed by a pointwise nonlinearity, such as

$$h_t = \psi(h_{t-1}, x_t) = \tanh(Wx_t + Uh_{t-1} + b_h)$$

In addition, the RNN may also have outputs $(y_1, y_2, \ldots, y_T), y_t \in \mathbb{R}^o$ that can be calculated by using another nonlinear function $\phi$

$$y_t = \phi(h_t, x_t)$$

From this recursion, the recurrent neural network naturally models the conditional probability $P(y_t|x_1, \ldots, x_t)$.

One problem with standard RNN is that it is difficult for them to learn long term dependencies [2, 9], and therefore in practice more sophisticated function $\psi$ are often used to alleviate this problem. For example the long short term memory (LSTM) [10] is one widely used recursive unit that is designed to learn long term dependencies. A layer LSTM consists of three gates and one memory cell, the computation of LSTM is as following[1]:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4}$$
$$c_t = f_t \odot c_{t-1} + (1 - f_t) \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$
$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where $W$, $U$, and $b$ represents the corresponding input-to-hidden, hidden-to-hidden weights and biases respectively. $\sigma(\cdot)$ denotes the sigmoid function, and $\odot$ is the elementwise product.

Another problem when using RNN to solve sequence to sequence learning problem is that it is not clear what strategy to apply when the input and output sequence does not share the same length (i.e. for outputs we have $T'$ time steps, which may not equal to $T$), which is the typical setting for this type of tasks. Sutskever et al. [16] propose to use an auto-encoder type of strategy, where the input sequence is encoded to a fixed length vector by using the last hidden state of the recurrent neural network, and then decode the output sequence from the vector. In more detail, let input and output sequence have $T$ and $T'$ time steps, and $f_e$, $f_d$ denote the encoding and decoding functions respectively, then the model tries to learn $P(y_1, \ldots, y_{T'}|x_1, \ldots, x_T)$ by

$$s \triangleq f_e(x_1, \ldots, x_T) = h_T \tag{7}$$
$$y_t \triangleq f_d(s, y_1, \ldots, y_{t-1}) \tag{8}$$

where $f_e$ and $f_d$ are implemented using multi-layer LSTMs.

## 4 SPELLING CORRECTION AS A FOREIGN LANGUAGE

It is easy to see that spelling correction problem can be formulated as a sequence to sequence learning problem as mentioned in section 3. In this sense, it is very similar to a machine translation problem, where the input is the misspelled text and the output is the corresponding correct spellings. One challenge for this formulation is that unlike in machine translation problem, the vocabulary is large but still limited[2]. However, in spelling correction, the input vocabulary is potentially unbounded, which rules out the possibility of applying word based encoding for this problem. In addition, the large output vocabulary is a general challenge in neural network based machine translation models because of the large Softmax output matrix.

The input/output vocabulary problem can be solved by using a character based encoding scheme. Although it seems appropriate for encoding the input, this scheme puts unnecessary burden on the decoder, since for a correction the decoder need to learn the correct spelling of the word, word boundaries, and etc. We choose the byte pair encoding (BPE) scheme [15] that strikes the balance between too large output vocabulary and too much learning burden for decoders. In this scheme, the vocabulary is built by recursively merging most frequent pairs of strings starting from character, and the vocabulary size is controlled by the number of merging iterations.

As shown in papers [1], encoding the whole input string to a single fixed length vector is not optimal, since it may not reserve all the information that is required for a successful decoding. Therefore, we introduce the attention mechanism from Bahdanau et al.[1] into this model. Formally, the attention model calculates a context vector $c_i$ from the encoding states $h_1, \ldots, h_T$ and decoder state $s_{i-1}$by

$$c_i = \sum_{j=1}^{T} \lambda_{ij} h_j \tag{9}$$
$$\lambda_{ij} = \frac{\exp\{a_{ij}\}}{\sum_{k=1}^{T} \exp\{a_{ik}\}} \tag{10}$$
$$a_{ij} = \tanh(W_s s_{i-1} + W_h h_j + b) \tag{11}$$

where $W_s$, $W_h$ are the weight vector for alignment model, and $b$ denotes the bias.

Now we are ready to introduce the full model for spelling correction. The model takes a sequence of input (characters or BPE encoded sub-words) $x_1, \ldots, x_T$ and outputs a sequence of BPE encoded sub-words $y_1, \ldots, y_{T'}$. For each input token the encoder learns a function $f_e$ to map to its hidden representation $h_t$

$$h_t = f_e(h_{t-1}, x_t; \theta_e) \tag{12}$$
$$h_0 = \mathbf{0} \tag{13}$$

The attentional decoder first obtain the context vector $c_t$ based on equation 10, and then learns a function $f_d$ that decodes $y_t$ from the context vector $c_t$

$$p(y_t|s_t) = \text{softmax}(Ws_t + b_d) \tag{14}$$
$$s_t = f_d(s_{t-1}, c_t; \theta_d) \tag{15}$$
$$s_0 = Uh_T \tag{16}$$

where $W$ and $b_d$ are the output matrix and bias, $U$ is a matrix that make sure that the hidden states of encoder would be consistent with the decoder's. In our implementation, both $f_e$ and $f_d$ are

[1] Sometimes additional weight matrix and vector are added to generate output from $h_t$ for LSTM, we choose to stick with the original formulation for simplicity.

[2] The vocabulary is limited in a sense that the number of words are upper bounded, in general

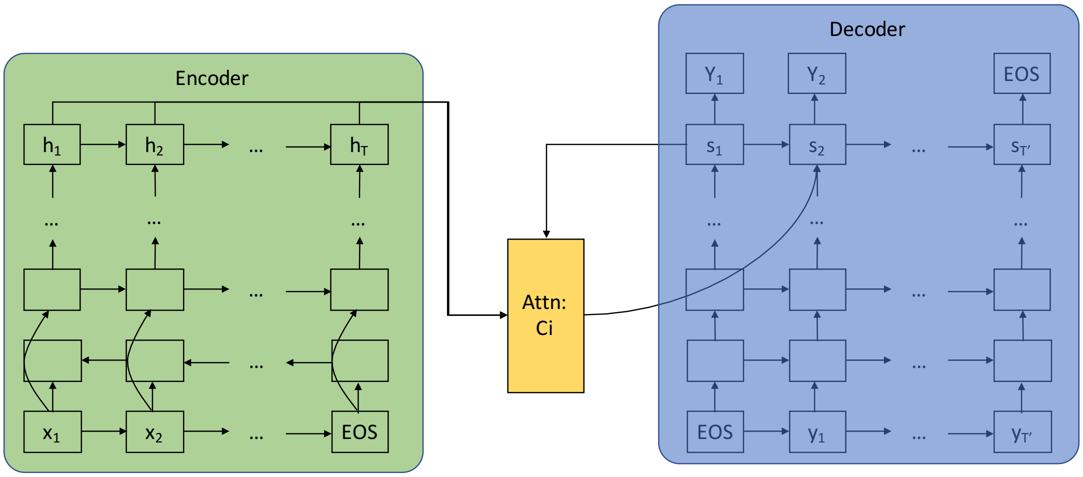

**Figure 1: Encoder-Decoder with attention framework used for spelling correction. The encoder is a multi-layer recurrent neural network, the first layer of encoder is a bidirectional recurrent neural network. The attention model produces a context vector $C_i$ based on all encoding hidden states $h_i$ and previous decoding state $s_{i-1}$. The decoder is a multi-layer recurrent neural network, and the decoding output $Y_i$ depend both on the context vector $c_i$ and the previous inputs $y_1 \dots y_{i-1}$.**

modeled using a multi-layer LSTM. As a whole, the end-to-end model is then trying to learn

$$p(y_1, \dots, y_{T'}|x_1, \dots, x_T) = \prod_{i=1}^{T'} p(y_i|x_1, \dots, x_T, y_{i-1}) \tag{17}$$

$$= \prod_{i=1}^{T'} p(y_i|f_d(s_{i-1}, c_i); \theta_d) \tag{18}$$

notice that in equation 18 the context vector $c_i$ is a function of the encoding function $f_e$, so we are not left the encoder isolated. Since all components are smooth and differentiable, the model can be easily trained with gradient based method to maximize the likelihood on the dataset.

## 5 EXPERIMENTS

We test our model in the setting of correcting e-commerce queries. Unlike machine translation problem, there is no public datasets for e-commerce spelling correction, and therefore we collect both training and evaluation data internally. For training data, we use the event logs that tracks user behavior on an e-commerce website. Our heuristic for finding potential spelling related queries is based on consecutive user actions in one search session. The hypothesis is that users will try to modify the search query until the search result is desirable with the search intent, and from this sequence of action on queries we can potentially extract the misspelling and correct spelled query pair. Obviously, this includes a lot more diversity on query activities besides spelling mistakes, and thus additional filtering is required to obtain representative data for spelling correction. We use the same techniques as Hasan et al.[8]. Filtering multiple months of data from our data warehouse, we got about 70 million misspelling and spell correction pairs as our training data. For testing, we use the same dataset as in paper [8], where it contains 4602 queries and the samples are labeled by human.

**Table 1: Results on test dataset with various methods. C-2-C denotes that the model uses character based encoder and decoder; W-2-W denotes that the model uses BPE partial word based encoder and decoder; and C-2-W denotes that the model uses a character based encoder and BPE partial word based decoder.**

| Method | Accuracy |
|---|---|
| Hasan et al.[8] | 62.0% |
| C-2-W RNN | 59.9 % |
| W-2-W RNN | 62.5 % |
| C-2-C RNN | 55.1% |

We use beam search to obtain the final result from the model. The result is illustrated in table 1, it is clear that our albeit much simpler, our RNN based model offers competitive performance as compare to the previous methods. It is interesting to note that, the BPE based encoder and decoder performs the best. The better performance may attribute to the shorter resultant sequence as compared to the character case, and possibly more semantic meaningful segments from the sub-words as compared to the characters. Surprisingly, the character based decoder performs quite well considering the complexity of the learning task. This demonstrated the benefit from end-to-end training and the robustness of the framework.

## 6 CONCLUSION

In this paper, we reformulated the spelling correction problem as a machine translation task under the encoder-decoder framework. The reformulation allowed us to use a single model for solving the problem and can be trained from end-to-end. We demonstrate the effectiveness of this model using an internal dataset, where the training data is automatically obtained from user logs. Despite the simplicity of the model, it performed competitively as compared to the state of the art methods that require a lot of feature engineering and human intervention.

## REFERENCES

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

[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[3] Eric Brill and Robert C Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 286–293.

[4] Silviu Cucerzan and Eric Brill. 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 293–300.

[5] Steffen Eger, Tim vor der Brück, and Alexander Mehler. 2016. A comparison of four character-level string-to-string translation models for (OCR) spelling error correction. *The Prague Bulletin of Mathematical Linguistics* 105, 1 (2016), 77–99.

[6] Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. 2010. A large scale ranker-based system for search query spelling correction. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 358–366.

[7] Jai Gupta, Zhen Qin, Michael Bendersky, and Donald Metzler. 2019. Personalized Online Spell Correction for Personal Search. In *The World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 2785–2791. https://doi.org/10.1145/3308558.3313706

[8] Sasa Hasan, Carmen Heger, and Saab Mansour. 2015. Spelling Correction of User Search Queries through Statistical Machine Translation.. In *EMNLP*. 451–460.

[9] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Mark D Kernighan, Kenneth W Church, and William A Gale. 1990. A spelling correction program based on a noisy channel model. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 205–210.

[12] Yanen Li, Huizhong Duan, and ChengXiang Zhai. 2012. CloudSpeller: query spelling correction by using a unified hidden markov model with web-scale resources. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 561–562.

[13] Chris J Lu, Alan R Aronson, Sonya E Shooshan, and Dina Demner-Fushman. 2019. Spell checker for consumer language (CSpell). *Journal of the American Medical Informatics Association* 26, 3 (01 2019), 211–218. https://doi.org/10.1093/jamia/ocy171 arXiv:http://oup.prod.sis.lan/jamia/article-pdf/26/3/211/27642469/ocy171.pdf

[14] Stephan Raaijmakers. 2013. A deep graphical model for spelling correction. (2013).

[15] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* (2015).

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[17] Casey Whitelaw, Ben Hutchinson, Grace Y Chung, and Gerard Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 890–899.