Automatic Query Correction for POI Retrieval using Deep and Statistical Collaborative Model

Canxiang Zhu, **Zhiming Chen**, **Yang Liu**, **Juan Hu**, **Shujuan Sun***, **Bixiao Cheng**, **Zhendong Yang**, **Li Ma** and **Hua Chai**
Didi Chuxing, Beijing, China
{Canxiang Zhu, Zhiming Chen, Yang Liu, Juan Hu, Shujuan Sun, Bixiao Cheng, Zhendong Yang, Li Ma, Hua Chai}@didiglobal.com

Abstract

Automatic text correction is a well-known and challenging task in Natural Language Processing. Query correction for Chinese POI(Point of Interests), especially, is a question that quite sophisticated and lack of research before. In previous works, deep learning has been employed in query correction with fair performance. It shows superior power of capturing spelling errors, but has the problem of miscorrecting texts that are correct, which is named as overcorrection. In this paper, we propose a character based sequence-to-sequence(Seq2Seq) model integrated with the Hidden Markov Model(HMM) named as HMM-Seq2Seq Model for POI Correction(HSMPC). Our model develops a character-level Long-Short-Term-Memory(LSTM) encoder to capture the error representations, and combines LSTM attention neural networks with HMM statistical model as decoder to refine the queries. We train the model on data automatically extracted from the search logs and POI data. The method is applied to intelligent transportation field to correct user entered query when selecting departure or destination. We perform experiments on the test corpus which is obtained from search logs and is annotated by human, the results demonstrate that our proposed model outperforms the baseline Moses statistical machine translation method, HMM and other deep learning models in terms of accuracy.

1 Introduction

Text or spelling correction plays an important role in many Internet interactive services such as search engines, e-commerce and intelligent transportation. The service analyzes the input entered by user to give the desired result. It is common for the user to make misspelled query, so many websites offer text autocorrection in the form of "Do you mean..." or "What you need is ..." suggestions to satisfy the users. Spelling correction is not a negligible task in intelligent transportation service. In scene of mobile service, misspellings are more likely to occur, which is caused by entering short and misleading content query in mobile keyboards. Without text correction, the service could easily fail to retrieve departure locations and destinations.

Spelling correction has been a topic for research for many years and numerous works have been conducted on it. Some probabilistic models were proprosed first. [Damerau, 1964] proposed a method that calculates a probabilistic edit distance for error detection and correction. [Mays *et al.*, 1991; Brill and Moore, 2000] applied the noisy channel model which is consisted of a source model and a channel model for spelling correction. [Mays *et al.*, 1991] used words bigram for the source model, and described the relation between the error word and its correction by defining four edit operations. [Brill and Moore, 2000] presented an improved error probabilistic model which estimates editing probabilities on misspelled-correct word pairs extracted from large amount of websites. [Ahmad and Kondrak, 2005] introduced the Expectation Maximization algorithm to learn edit distance weights directly from search query logs.

Spelling correction can be described as transforming one text to another. Researchers have introduced machine translation ideas to capture some context information for improving the performance compared to rule based models. [Hasan *et al.*, 2015] used character-based statistical machine translation(SMT) to correct user input query in the e-commerce domain. This method extracted training data from event logs that record the user interactions, and the open source tool Moses was applied to extract phrase table. [Liu *et al.*, 2013] tackled this problem as three components: a word segmentation based language model to generate correction candidates; a SMT model to provide correction candidates and a Support Vector Machine(SVM) classifier to rerank the candidates provided by the previous two components. [Sun *et al.*, 2010] also assumed that query spelling correction is performed at the phrase level. The mutual weaknesses of the SMTs lie in capturing deep semantics and long dependencies.

Some works adopted deep learning models for spelling correction recently. [Sun *et al.*, 2015] proposed a convolutional neural networks(CNN) for text correction.[Ghosh and Kristensson, 2017] developed a character level CNN and gated recurrent unit(GRU) encoder along with a word level GRU attention decoder for text correction and completion in keyboard decoding.[Etoori *et al.*, 2018] proposed a char-

*Contact Author

acter based Seq2Seq text Correction Model for Indic Languages(SCMIL). [Li *et al.*, 2018] came up with a two-stage hybrid system which combined the BiLSTM-CRF model for detection and three Grammatical Error Correction(GEC) models for correction.

In summary, deep learning methods have advantages in capturing spelling error information from abundant data, but have overcorrection problem. Statistical methods are more faithful to the original text, while lacking ability of representing the context information of query errors. In this paper, we propose a sequence-to-sequence model which combining deep learning and a statistical method. Our main contributions in this paper are summarized as follows:

- We present a character based Seq2Seq model integrated with HMM for POI query correction.
- We analyze the characteristics of POI query error, which has been taken into account in our model.
- We create the training corpus automatically extracted from the query logs, and build the test dataset. We describe the steps of data processing in section 4.
- We compare the performance of the proposed model with Moses, HMM and other SOTA Seq2Seq models on test set.

2 Background

2.1 Sequence-to-Sequence Model

Sequence-to-Sequence(Seq2Seq) model [Sutskever *et al.*, 2014; Cho *et al.*, 2014] is composed of an encoder and a decoder. [Bahdanau *et al.*, 2014] moves a step further and implements a mechanism of attention in the decoder. The encoder transforms a source sentence $\mathbf{x} = (x_1, x_2, x_3, ..., x_n)$ into context vectors $\mathbf{c} = (c_1, c_2, c_3, ..., c_n)$ using stacked Long Short Term Memory(LSTM) layers [Hochreiter and Schmidhuber, 1997] or Gated Recurrent Units (GRU) layers [Chung *et al.*, 2014]. x_i is the embedding of the i-th word in the source sentence. h_i is the hidden state of the top layer in encoder. The decoder generates the next word by the given context vector and all the previously predicted words. In a word, the decoder calculates a probability over the target sentence by decomposing the joint probability into the orderd conditional probabilities:

$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t | y_1, \cdots, y_{t-1}, \mathbf{c}) \tag{1}$$

where $\mathbf{y} = \{y_1, y_2, \cdots, y_T\}$ is the output target sequence.

2.2 Hidden Markov Model

Hidden Markov model is an effective way to model language sequences by assuming the process which generates symbols satisfies Markov property. Specifically, HMMs presume observed tokens in a sentence are emitted by a latent variable(hidden state). According to Markov property, the current hidden state is only effected by the previous hidden state. Formally, this gives us an emission $p(x_t|z_t)$ and transition $p(z_t|z_{t-1})$ probability. The graphical model is drawn in Figure 1. Where x_t depict the observations and z_t are the latent variables. Now we can use the emission and transition probability to calculate the joint probability distribution(Eq.2).

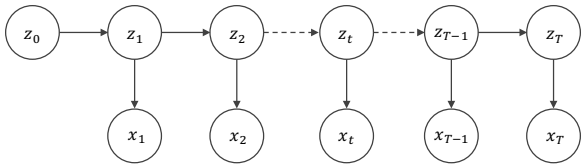


Figure 1: Structure of Hidden Markov Model. $z_t(t = 1, \ldots, T)$ denotes the latent state which only depends on the previous state z_{t-1}, and emits an observation $x_t(t = 1, \ldots, T)$ at each time step. z_0 is the initial state.

$$p(x_1, ..., x_T, z_1, ..., z_T) = p(z_0) \prod_{t=1}^{t=T} p(z_t|z_{t-1}) \prod_{t=1}^{t=T} p(x_t|z_t) \tag{2}$$

Hidden Markov Model is composed of an initial state z_0 and emission probability $p(x_t|z_t)$ and transition probability $p(z_t|z_{t-1})$. The hidden state is controlled by initial state and transition probability. The observation state is controlled by emission probability [Hassan and Nath, 2005].

3 Integrating HMM into Seq2Seq

In this section, we first present properties of query correction, and discuss the aspects in which HMM and Seq2Seq are more qualified. Then the collaboration of HMM and Seq2Seq designed in our work is illustrated in detail.

3.1 Features of Chinese Query Correction

It is noteworthy that, query correction quite differs from normal circumstances of text correction, due to the fact that query correction includes the complex combination of features from Chinese characters and POIs.

3.1.1 Chineses Characters Related Correction

The most common manner to type a Chinese character is using a pinyin method. Pinyin is the way to pronounce and spell Chinese characters written with the Latin alphabet. In the meanwhile, Chinese characters are composed of calligraphic strokes which can be classified as a set of line patterns. Users can type or draw these line patterns to input Chinese characters too. The diverse expressions bring on sophisticate error types. For clarity, We divide Chinese queries into the following three types:

Right query(RQ). RQ refers to the query equals with correct query, examples show as row 6, 8, 10 in Table 1.

Normal spelling error query(NSEQ). NSEQ portrays queries which are close in Chinese morphology, including pinyin and the glyph of Chinese characters. Similar pinyin can be divided into front nasal, back nasal, flat tongue, curly tongue, etc, "san, shan, shang, sang" for example. Similar glyph mainly contains same radical, same stokes with different order, similar stokes, etc, "峰,锋,木,本,晃,旮,己,已,巳" for example. Levenshtein edit distance is a common method to measure the distance between two text segments, denoted by $D(s, c)$.

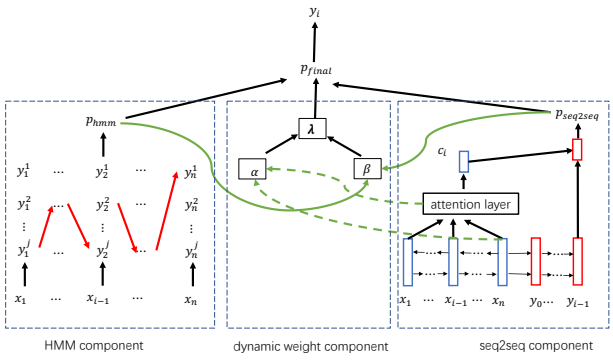


Figure 2: Structure of HSMPC. $x_i (i = 1, \ldots, n)$ denotes the input word embeddings. y_i denotes the output word embeddings. p_{hmm}, $p_{Seq2Seq}$ and p_{final} are the output probabilities of HMM, Seq2Seq model and HSMPC. y_i^j is the i-th word's hidden state for the j-th word in the target vocabulary. α, β and λ are context weight, confidence weight and the weight combining the previous two.

In this paper, we calculate three forms of edit distance and set $D(s, c)$ to the minimum value. (1) The default Levenshtein calculation method operating on characters which are not exactly matched; (2) Transfer s and c to their pinyin forms, and two Chinese characters from s and c, denoted by w_s and c_s, would be considered as a matching if their pinyin forms are identical; (3) Continue to relax matching restrictions to condition that w_s and c_s have similar glyph. Correction pairs satisfied $D(s, c) = 0$ are classified as NSEQ.

Special spelling error query(SSEQ). SSEQ is query that does not belong to RQ or NSEQ, in other words, SSEQ obtains a relatively large edit distance for correct query, such as query with continual pinyin and inverted words.

3.1.2 Features of POI Related Correction

Expressions of a POI vary in different perspectives. Normally, it is described as a name or address. When it lies in a map, location semantic becomes a key property to discriminate from other POIs.

Location information. Corrections of queries are strongly related to the city in which corresponding POI located. Normally, roads or communities incline to be named with features that is owned specifically by each city. That means even for same wrong query, the corrections are diverse in different citites.

Sparsity of POI data. In transportation domain, query needs of POIs differ extremely from their locations. POIs located in urban area tend to be requested more frequently than those in suburb. Queries for remote area are sparse in our dataset.

3.2 HSMPC Model

The HMM works pretty well on RQ and NSEQ, but do not work for SSEQ. In contrast, Seq2Seq model outperforms the HMM on SSEQ, but not remarkable as HMM on RQ. These conclusions are proved by our experiemnts in section 5.3. As far as our concerned, it may due to the fact that the HMM mainly learns from POI data which are valid and regular, while the Seq2Seq model learns from user search logs whose validity is not guaranteed and full of noise. In this regard, we suppose that integrating HMM with Seq2Seq model may take full advantage of the two models and obtains better performance.

The architecture of our integrated model is illustrated in Figure 2. It involves three components: the HMM, the Seq2Seq model and the dynamic weight. The outputs of the two components are combined with the dynamic weight to produce final consolidated results. It is worthy to compendiously introduce the implement of HMM for spelling correction. The observation states are derived from user inputted query using bigram words. The hidden states are recalled by pinyin taking pronunciation, NPLIM and polyphones into account. The transition probability is calculated by counting bigram words from POI data. The emission probability is measured by rules such as edit distance of Chinese characters and pinyin. Especially, the hidden state which is indentical to user's input word aquires an emission probability of 1.

For the encoder of Seq2Seq model, a bidirectional LSTM layer is adopted to output context vectors **c**. For the decoder, the output probabilities for each word is computed with a softmax function, equated:

$$p_{Seq2Seq} = softmax(logit_{Seq2Seq}) \tag{3}$$

where $logit_{Seq2Seq}$ is the output of the decoder, computed by

$$logit_{Seq2Seq} = f(s_{j-1}, y_{j-1}, c_j) \tag{4}$$

f is a nonlinear function usually adopts $tanh$, s_{j-1} is the previous hidden state, y_{j-1} is the current word embedding, c_j is the context vector. We modify the $p_{Seq2Seq}$ by combining $p_{Seq2Seq}$ with p_{hmm} as follows:

$$p_{final} = \lambda \cdot p_{Seq2Seq} + (1 - \lambda) \cdot p_{hmm} \tag{5}$$

where λ is the dynamic weight to adjust the contribution from the HMM and Seq2Seq model, p_{hmm} is the probability output by HMM, computed by

$$p_{hmm} = g(p_{\mathbf{y}}, p^{t-1}_{transit}, p^{t}_{emit}) \tag{6}$$

$p_{\mathbf{y}}$ is the jointed probability of the output words $\mathbf{y} = (y_1, y_2, \cdots, y_{t-1})$, $p^{t-1}_{transit}$ is the transition probability from y_{t-1} to y_t, p^{t}_{emit} is the emission probability from hidden state y_t to observation state o_t, g is a multiplication function.

It is quite worth to be mentioned that we meticulously design the dynamic weight λ. We mainly consider the following two factors: the context of the current word, the distribution of the predicted probability both for the Seq2Seq model and HMM. Our dynamic weight is computed by

$$\lambda = sigmoid(\gamma \cdot \alpha + (1 - \gamma) \cdot \beta) \tag{7}$$

where α is the weight reflects the context of the current word, β is the weight reflects the distribution of the predicted probability, which exhibits the predict confidences of the two models. γ is the weight reflects the contribution from the α and β. α is calculated as follows:

$$\alpha = sigmoid(tanh(W_1 \cdot s_{j-1}) + tanh(W_2 \cdot c_j)) \tag{8}$$

Row	Query	Correct query	HMM result	Seq2Seq result	HSMPC(λ) result	Best Model	Query type	Result type
1	“双桥夜”	“双桥路”	“双桥夜”	“双桥街”	“双桥路”	HSMPC	SSEQ	G
2	“wuyicun”	“五一村”	“wuyicun”	“武夷村”	“五一村”	HSMPC	SSEQ	G
3	“施甸县委”	“施甸县甸”	“施甸县委”	“施甸县文”	“施甸县甸”	HSMPC	SSEQ	G
4	“丰贤路9”	“奉贤路9”	“丰翔路9”	“丰贤路9”	“奉贤路9”	HSMPC	NSEQ	G
5	“老平江”	“号平江”	“老平江”	“老平江”	“号平江”	HSMPC	SSEQ	G
6	“晋公桥”	“晋公桥”	“晋公桥”	“金公桥”	“晋公桥”	HMM and HSMPC	RQ	G
7	“爱家宠物”	“爱佳宠物”	“爱佳宠物”	“爱家宠物”	“爱佳宠物”	HMM and HSMPC	NSEQ	G
8	“圆润”	“圆润”	“圆润”	“源润”	“圆润”	HMM and HSMPC	RQ	G
9	“山福”	“三福”	“三福”	“千福”	“三福”	HMM and HSMPC	NSEQ	G
10	“胜利美”	“胜利美”	“胜利美”	“胜利街”	“胜利美”	HMM and HSMPC	RQ	G
11	“池东工业区”	“慈东工业区”	“慈东工业区”	“池东工业区”	“池东工业园”	HMM	NSEQ	S
12	“丽思花园南门”	“丽斯花园南门”	“丽斯花园南门”	“丽思花园南门”	“丽思花园南门”	HMM	NSEQ	S
13	“广阜屯”	“广埠屯”	“广埠屯”	“广埠屯”	“广埠屯”	HMM, Seq2Seq and HSMPC	NSEQ	S
14	“广交会c”	“广交会展”	“广交会c”	“广交会展”	“广交会展”	Seq2Seq and HSMPC	SSEQ	S
15	“刘xuan”	“刘园”	“刘xuan”	“刘园”	“刘园”	Seq2Seq and HSMPC	NSEQ	S
16	“服保原”	“福保园”	“妇保院”	“福保园”	“福保苑”	Seq2Seq	SSEQ	B
17	“湖畔厅”	“湖畔庭”	“湖畔亭”	“湖畔庭”	“湖畔新”	Seq2Seq	NSEQ	B
18	“信大思”	“信达思”	“信达思”	“信达思”	“信达斯”	HMM and Seq2Seq	NSEQ	B

Table 1: Correction examples of Seq2Seq, HMM and HSMPC, which illustrate the different performance on query correction types.

W_1 and W_2 are learnable parameters. β is calculated by

$$\beta = sigmoid(\frac{\mu \cdot E_{hmm}}{\mu \cdot E_{hmm} + E_{Seq2Seq}}) \tag{9}$$

where μ is the parameter to adjust the difference on magnitude for the predicted probabilities of the two models, E_{hmm} is entropy of the predicted probability for HMM, meanwhile $E_{Seq2Seq}$ is entropy for Seq2Seq model. E_{hmm} and $E_{Seq2Seq}$ can be calculated as follows:

$$E_{hmm} = -\sum(p_{hmm} \cdot log(p_{hmm})) \tag{10}$$

$$E_{Seq2Seq} = -\sum(p_{Seq2Seq} \cdot log(p_{Seq2Seq})) \tag{11}$$

α, β, γ, μ W_1 and W_2 are learnable parameters. α, β, W_1, W_2 are multi-dimensional matrixs, γ, while μ are scalars. We denote the model with a dynamic weight of λ as HSMPC(λ). We can also replace the λ in equation 5 with α or β denote as HSMPC(α) or HSMPC(β).

4 Data Processing

Users input queries in the searching bar to select the departure locations or destinations. Users change queries to re-search if wanted POI is not existed in retrieved POI list. These information can help us greatly to improve the user experience. All queries typed in the searching bar before a user actually clicking a retrieved POI are kept tracking by an event log. These queries in the event log constitute a session. Table 2 summarizes examples of session logs. We first illustrate the extraction of session logs, then filtering methods are applied to prune correction pairs introducing invalid modifications. In the end, we refine the correction pairs by using information of POI.

4.1 Extraction of Sessions

The idea of extracting correction training data from session logs is based on an assumption that users change query because they typed the wrong query to search, and try to modify the query until the satisfied POI is listed. We express the final queries leading to clicking on retrieved POI list as order queries. And the source queries and order queries form the correction pairs together, denoted by (s, c). s is the source query and c is the order query, i.e. the correction.

Session	Correction pair	POI name
“北”, “北京”, “北京邮电”	(“北京邮电”, “北京邮电”)	“北京邮电大学”
“第一人民医院”, “普陀中心医院”	-	“上海市普陀中心医院”
..., “中美宜和”	-	-

Table 2: Example of session logs. The correction pairs are extracted from session logs.

It is noteworthy that, modifications are not always indicating valid corrections of mistyped words or characters. Table 2 exhibits three other common situations respectively: (1) Adding words to complete information about the desired POI; and (2) User changes their intention to select another location; (3) User's correction doesn't retrieve satisfied results. The first kind of sessions are still useful, can help models learn which queries do not need to be corrected, i.e. $s = c$. The second sessions should be discarded since the alteration are personalized. As to the last circumstance, we have to drop these sessions as lacking of persuasiveness.

All sessions are collected day by day. Months of session logs are produced to be training dataset. For the assurance of keeping test dataset unseen to models, the test dataset is extracted from posterior days with same process.

4.2 Filtering and Refinements for Correction Pairs

After extracted from reduced sessions, these rough correction pairs still contain noises. A series of procedures are utilized to improve the quality of correction pairs.

We first employ some regular text pre-processing techniques to normalize correction pairs, for instance to drop punctuations and transfer capital letters to lowercase. And correction pairs (s, c) would be excluded under the following circumstances:

(1) If s is substring of c or c is substring of s. The operations do not introduce useful corrections because they are just inserting or deleting information.

(2) If the altering between s and c is just number or geographical regions substitution.

(3) If the relaxed edit distance described above between s and c exceeds a threshold. Empirically we set the edit distance threshold to 2, resulting effective reducing of intention shift related modifications.

As mentioned before, corrections sometimes contain errors too. We utilize the information contained in POI to refine the corrections to improve the quality of training data.

Transfer both correction and corresponding POI to pinyin. If the pinyin of correction is the substring of POI's pinyin, the correction would be replaced by the matched substring of POI entirely. If no matching is found, we search correction string directly in the POI with toleration of similar glyph. If two Chinese characters have similar glyph, they would be considered as a matching.

For further completion of information from POI, city names are inserted at the head of s and c. Corrections of queries are strongly related to the city at which corresponding POI located. Normally, roads or communities tend to be named with features that owned specifically by each city.

The rules of omitting correction pairs are relatively rough and aggressive, might neglect some illuminating cases and bring flaws to training dataset. The methods used to refine the corrections pairs are rather naive and inflexible. We will introduce more advanced distance measures and similarity models to help improve the quality of corrections pairs in the future.

5 Experiments

In this section, we introduce the experimental details, including the dataset, the experiment settings, experiment results and analysis.

5.1 Datasets

After the processing of data described in section 4, we gain large amount of preprocessed correction pairs. We build our training set with 5 million correction pairs for Moses, Seq2Seq and our proposed models, which contains queries that need corrections or not with ratio of 5:2 set empirically. User inputted query plays a role as source sentences, while the correct query is denoted as the target counterpart. The name of user's city is regarded as a single word and is added at the first place of the inputted query and the correct query. All correction pairs in datasets are character granularity. The average length of the inputted query and correct query is 4.38. The developing set is consited of 6000 query pairs. The test set contains 2000 query pairs including 300 query pairs that not need be corrected. All of the query pairs in develping set and test set are labeled by human, to ensure the accuracy of evaluation. Table 3 shows a couple of samples of the our datasets. For HMM model, the training set is derived from our POI database with more than 60 million POI data. We mainly use POI's name as our training data, such as "西直门", "仙云庄" and "北京西站".

5.2 Setup

In this paper, we train four types of models: Moses, HMM model, Seq2Seq model and the proposed models, which are referred as HSMPC. For HMM model, we prune candidate target words according to the probability conducted from equation 6, which can reduce computational complexity of viterbi process [Forney, 1973]. For Seq2Seq model, we assign the hyper-parameters as follows: the word embedding size as 256, the hidden units number to 512, the batch size as 1024. And limit the maximum sentence length as 50. Adadelta [Zeiler, 2012] is adopted to optimize the Seq2Seq model. The early stop patience is set to 10. During the inference, we set beam size to 3. After adding hmm score to Seq2Seq, normalization layer is appended. We implement Seq2Seq and HSMPC model by Tensorflow [Abadi *et al.*, 2016]. All experiments are performed on a NVIDIA Tesla P40 GPU. For Moses system, we adopt 8 commonly used features. These features have been proved to improve translation quality effectively, which contains the bidirectional translation probabilities, the bidirectional lexical weights, the language model, the reordering model, the word penalty and the phrase penalty.

Row	Input query	Correct query
1	"南京市 清 山 小 区"	"南京市 青 山 小 区"
2	"江门市 江 州 中 学"	"江门市 江 洲 中 学"
3	"沈阳市 大 唐"	"沈阳市 大 唐"

Table 3: Example of datasets.

Model	TOP1	TOP3	Overcorrection
Moses(baseline)	60.05%	84.30%	10.33%
HMM(baseline)	62.30%	87.05%	**6.67%**
Seq2Seq(baseline)	75.00%	88.45%	9.33%
HSMPC(α)	76.25%	89.20%	7.67%
HSMPC(β)	75.55%	89.10%	7.67%
HSMPC(λ)	**76.50%**	**89.55%**	7.33%

Table 4: The performances of Moses, HMM, Seq2Seq, HSMPC(α), HSMPC(β) and HSMPC(λ) on test set. The overcorrection rate denotes the error correction rate on right queries.

5.3 Results

We conduct experiments with six models: Moses, HMM and Seq2Seq as baseline models, HSMPC(α) , HSMPC(β) and HSMPC(λ) are our proposed models. We mainly pay attention to the performances of top1 accuracy, top3 accuracy and overcorrection rate for the six models. The experiment results on test set shows in Table 4. Moses obtains the worst performance on the three metrics. The HMM outperforms other five models on overcorrection rate. The Seq2Seq model exceeds HMM 12.7% score in terms of top accuracy and 1.40% score for top3 accuracy, but worse than HMM for overcorrection rate with 3.25% score. Joyfully, our three integrated models obtain better scores. HSMPC(α) improves the baseline Seq2Seq model with 1.25%, 0.75% score for top1 and top3 accuracy and decrease 1.67% score for overcorrection rate. For HSMPC(β), the improving scores are 0.55%, 0.65%, 1.67%, while 1.50%, 1.10% and 2.00% for HSMPC(λ). We can obtain the following conclusions: 1) Neural network mdoels performance better than Mose and

Model	RQ	NSEQ	SSEQ
Moses(baseline)	88.48%	72.67%	10.13%
HMM(baseline)	**93.32%**	75.50%	5.19%
Seq2Seq(baseline)	90.27%	80.41%	52.76%
HSMPC(α)	92.33%	84.58%	52.17%
HSMPC(β)	92.41%	84.48%	50.22%
HSMPC(λ)	92.61%	**86.72%**	**53.32%**

Table 5: The performances of Moses, HMM, Seq2Seq, HSMPC(α), HSMPC(β) and HSMPC(λ) for diffirent types on 200 samples.

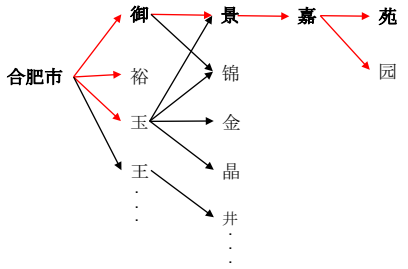
HMM. 2) Our proposed models obtain notable improvements comparing to Seq2Seq. 3) Both the dynamic weight α and β work well, integrating α and β behaves better.

5.4 Analysis

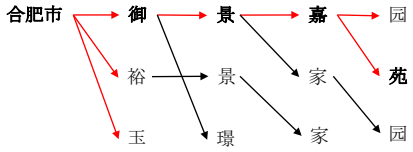
In the above section, we demonstrate the integrated model outperforming Seq2Seq and HMM. In this part, for the purpose of understanding how our method works, we analyse the experiment results by going into details, mainly from the follow two viewpoints of the integrated model: the behavior on query correction and the internal workflow.

5.4.1 Behavior of integrated model

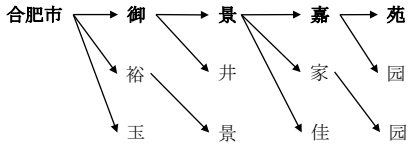
In section 3, we demonstrate the characteristics of HMM and Seq2Seq on the basis of experiment results HMM and Seq2Seq. The experiment results showing in section 5.3 proves our proposed model conducts better query correction. We are quite curious about whether the performance aquired by our proposed model is owing to full advantages of HMM and Seq2Seq being taken. Apart from comparing the accuracy and overcorrection rate on test set, we meticulously analyse 200 queries choosing from test set randomly. The 200 sample queries is composed with three types of queries(RQ, NSEQ, SSEQ) with ratios 16.5%, 62.0% and 21.5% seperately. Table 5 shows the accuracy of different models on different type queries. HMM obtains the highest score on RQ, while integrated models aquires scores between HMM and Seq2Seq, indicating HMM helps improve the Seq2Seq's performance on RQ. In other words, integrating HMM to Seq2Seq reduces overcorrection. For NSEQ and SSEQ, integrated model outperform HMM and Seq2Seq. Table 1 are some correcting cases.The last column refer to the effect type of integrating HMM into Seq2Seq. In other words, it indicates whether integrated model better than Seq2Seq. We categorize effect type to three: good effect(G), no effect or unuseful effect(S), bad effect(B). For the 200 sample queries, the numbers of G, S and B are 16, 174 and 10, namely the ratios are 8%, 87% and 5%. With these analysis, we can conduct the conclusion that our proposed model successfully takes advantages of HMM and Seq2Seq and outperforms the two models. Furthermore, HSMPC model can correct error reated to city or geographical location. For an input query "li水桥" requested in Beijing and Shanghai separately, our model gives the right result "立水桥" in Beijing and "李水桥" in Shanghai. All the three baseline models fail to produce the right corrections, and gives the same result "立水桥" instead, which does not exist in Shanghai.



(a) The viterbi decoding process of the HMM.



(b) The beam search decoding process of the Seq2Seq.



(c) The beam search decoding process of the HSMPC

Figure 3: The inference processes of three models: (a) HMM, (b) Seq2Seq, (c) HSMPC. The red arrow represent the mutual decoding path of HMM and Seq2Seq. There are 3 candidate words at each decoding step in subgraph (b) and (c) steming from the setting of beam size.

5.4.2 Internal Workflow of HSMPC

We depict the decoding processes of one same query to illustrate the differences among all models as figure 3. The origin query is "合肥市 御 景 嘉 园", while the right correction should be "合肥市 御 景 嘉 苑", denoted in bold text in the figure. More specifically, the correcting operation needs to transfer the word "园" to "苑". Subfigure (a) shows the decoding process of HMM, which acquires the best result "合肥市 御 景 嘉 苑" successively. While in subfigure (b), the decoder of Seq2Seq produces "合肥市 御 景 嘉 园" as the highest confident result, though it is not correct. We picture the mutual decoding path with red arrows in figure 3. Take "嘉" as an example, the preceding words in the decoding path are the same, i.e. "合肥市 御 景". When both models generate the next word, the probability of the word "苑" scores highest in HMM while lower than the word "园" in Seq2Seq. We fuse these probabilities in the manner of equation 5 and the collaborated decoding process produces the right result ultimately which represented in subfigure (c).

6 Conclusions

We propose a HSMPC model which combines the advantages of Seq2Seq model and HMM for POI related query correction. Our work shows that it can significantly improve the performance comparing to baseline models.

References

[Abadi *et al.*, 2016] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[Ahmad and Kondrak, 2005] Farooq Ahmad and Grzegorz Kondrak. Learning a spelling error model from search query logs. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 955–962, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[Brill and Moore, 2000] Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 286–293, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[Chung *et al.*, 2014] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[Damerau, 1964] Fred J. Damerau. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176, March 1964.

[Etoori *et al.*, 2018] Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152. Association for Computational Linguistics, 2018.

[Forney, 1973] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[Ghosh and Kristensson, 2017] Shaona Ghosh and Per Ola Kristensson. Neural networks for text correction and completion in keyboard decoding. *CoRR*, abs/1709.06429, 2017.

[Hasan *et al.*, 2015] Saša Hasan, Carmen Heger, and Saab Mansour. Spelling correction of user search queries through statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 451–460, 2015.

[Hassan and Nath, 2005] Md. Rafiul Hassan and Baikunth Nath. Stockmarket forecasting using hidden markov model: A new approach. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, ISDA '05, pages 192–196, Washington, DC, USA, 2005. IEEE Computer Society.

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

[Li *et al.*, 2018] Chen Li, Junpei Zhou, Zuyi Bao, Hengyou Liu, Guangwei Xu, and Linlin Li. A hybrid system for chinese grammatical error diagnosis and correction. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 60–69. Association for Computational Linguistics, 2018.

[Liu *et al.*, 2013] Xiaodong Liu, Kevin Cheng, Yanyan Luo, Kevin Duh, and Yuji Matsumoto. A hybrid chinese spelling correction using language model and statistical machine translation with reranking. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 54–58, 2013.

[Mays *et al.*, 1991] Eric Mays, Fred J. Damerau, and Robert L. Mercer. Context based spelling correction. *Inf. Process. Manage.*, 27(5):517–522, September 1991.

[Sun *et al.*, 2010] Xu Sun, Jianfeng Gao, Daniel Micol, and Chris Quirk. Learning phrase-based spelling error models from clickthrough data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 266–274. Association for Computational Linguistics, 2010.

[Sun *et al.*, 2015] Chengjie Sun, Xiaoqiang Jin, Lei Lin, Yuming Zhao, and Xiaolong Wang. Convolutional neural networks for correcting english article errors. In Juanzi Li, Heng Ji, Dongyan Zhao, and Yansong Feng, editors, *Natural Language Processing and Chinese Computing*, pages 102–110, Cham, 2015. Springer International Publishing.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[Zeiler, 2012] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.