Cogn Comput (2013) 5:234–242
DOI 10.1007/s12559-012-9183-y

Common Sense Knowledge for Handwritten Chinese Text Recognition

Qiu-Feng Wang · Erik Cambria · Cheng-Lin Liu · Amir Hussain

Received: 30 April 2012 / Accepted: 9 August 2012 / Published online: 23 August 2012

Abstract Compared to human intelligence, computers are far short of common sense knowledge which people normally acquire during the formative years of their lives. This paper investigates the effects of employing common sense knowledge as a new linguistic context in handwritten Chinese text recognition. Three methods are introduced to supplement the standard n-gram language model: embedding model, direct model, and an ensemble of these two. The embedding model uses semantic similarities from common sense knowledge to make the n-gram probabilities estimation more reliable, especially for the unseen n-grams in the training text corpus. The direct model, in turn, considers the linguistic context of the whole document to make up for the short context limit of the n-gram model. The three models are evaluated on a large unconstrained handwriting database, CASIA-HWDB, and the results show that the adoption of common sense knowledge yields improvements in recognition performance, despite the reduced concept list hereby employed.

Q.-F. Wang (✉) · C.-L. Liu
National Laboratory of Pattern Recognition,
Institute of Automation of Chinese Academy of Sciences,
Beijing 100190, People's Republic of China
e-mail: wangqf@nlpr.ia.ac.cn

C.-L. Liu
e-mail: liucl@nlpr.ia.ac.cn

E. Cambria
Temasek Laboratories, National University
of Singapore, Singapore 117411, Singapore
e-mail: cambria@nus.edu.sg

A. Hussain
Department of Computing Science and Mathematics,
University of Stirling, Stirling FK9 4LA, UK
e-mail: ahu@cs.stir.ac.uk

Keywords Common sense knowledge · Natural language processing · Linguistic context · n-gram · Handwritten Chinese text recognition

Introduction

Common sense knowledge spans a huge portion of human experience, encompassing knowledge about the spatial, physical, social, temporal, and psychological aspects of typical everyday life [1]. Compared to human intelligence, computers are far short of common sense knowledge, which people normally acquire during the formative years of their lives.

In order to make machines smarter, many common sense knowledge bases, for example, Cyc [2], ConceptNet [3], SenticNet [4], Freebase [5], and Isanette [6], are currently being developed and maintained. So far, such knowledge bases have been exploited for natural language processing (NLP) tasks, such as information retrieval [7], sentiment analysis [8], speech recognition [9], multimedia management [10], predictive text entry [11], and e-health applications [12]. However, to the best of our knowledge, there has been no prior study on integrating common sense knowledge in handwritten Chinese text recognition (HCTR).

In the near fifty years research of Chinese handwriting recognition, most works focused on isolated character recognition [13] and string recognition with very strong lexical constraints, such as legal amount recognition in bank checks [14] and address phrase recognition [15]. For general text recognition (one example of Chinese handwritten page is shown in Fig. 1), however, first works have been reported only in recent years [16–18]. Despite such

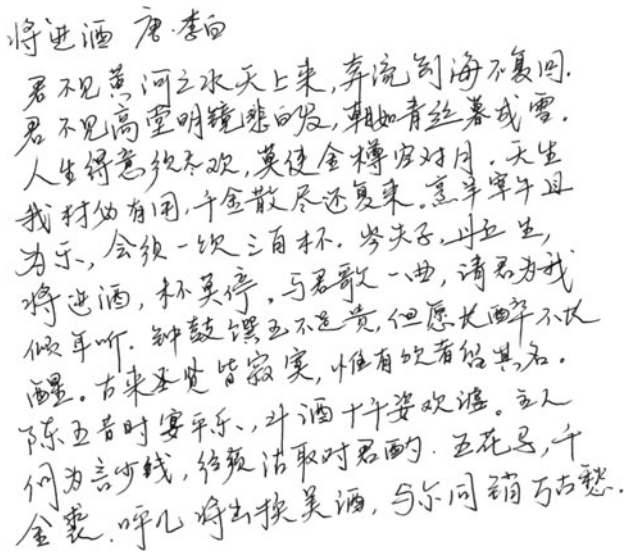

Fig. 1 A page of handwritten Chinese text

works show that linguistic context plays a very important role in HCTR, very simple language models are usually employed, even in the post-processing application [19]. The n-gram model, in particular, is the most popular for the ease of applicability and implementation. Such model, however, is a mere statistical method that does not exploit the semantics associated with natural language text [20].

The n-gram model, in fact, does not usually work well for unseen n-gram sequences in the training text corpus[1] and only considers a very short-distance context (for the moderate model size, n usually takes 2 or 3). Although many studies have been conducted to overcome these two limitations [20, 21], both have not been solved yet.

This paper reports our first attempt to exploit common sense knowledge for HCTR. The key idea is to use the semantic context and concept similarity provided by ConceptNet, to aid the n-gram model. In particular, we hereby propose three different approaches. Firstly, we embed the concept semantic similarity in the n-gram probability estimation, which is especially useful for those zero and low-frequency n-gram sequences. Secondly, we use the common sense knowledge as a direct linguistic context model to interpolate with the n-gram model (since this method calculates the similarity between current word and the whole document, it considers the long-distance context). Lastly, we combine such two methods to an ensemble model. We tested the proposed methods through several experiments on the CASIA-HWDB database [22]. Because we only used a reduced concept list (only 2.8 % of the normal word list in our system), the common sense knowledge yields a little improvement in recognition performance, but does show the potential of correcting recognition errors.

The rest of this paper is organized as follows: Related Works section reviews some related works about linguistic context in HCTR and other NLP tasks; System Overview section gives an overview of our HCTR system; Common Sense Knowledge section describes in detail the proposed approach for integrating common sense knowledge in HCTR; Experiments section presents experimental setup and results; Discussion section proposes a discussion on those results; Conclusion section, finally, draws some concluding remarks.

Related Works

There are several issues that need to be solved under the umbrella of HCTR, for example, character over-segmentation, character classification, geometric context, linguistic context, path evaluation and search, and parameters estimation. In this section, we briefly outline the importance of linguistic context in HCTR and other NLP tasks as it is the main issue of this paper. A more detailed state of the art on HCTR is available in a comprehensive work recently proposed by Wang et al. [18].

Linguistic context plays a very important role in HCTR, which can guide the system to choose the most likely sentence from the full set of candidates. However, early works on HCTR considered very limited linguistic context. A preliminary work by Su et al. [16] reported a very low correct rate of 39.37 % without any linguistic context. Later, several n-gram language models for linguistic context (e.g., character-level bi-gram, tri-gram, and word-level[2] bi-gram) were investigated in HCTR and improved the performance largely [17, 23]. To overcome the mismatch between language model and recognition task, language model adaptation was investigated in HCTR recently, resulting in a further improvement in the recognition performance [24].

Although there are no advanced language models used in HCTR at the present time, language modeling communities have extended n-gram models in many other NLP tasks [20, 21]. Most works focused on overcoming the limitation of short-distance context of n-gram model[3]. Skipping models use several distance n-gram models to approximate the higher order n-gram model [25]. Caching techniques exploit the fact that, after a word appears, it is

[1] In the case of unconstrained texts, no corpus is wide enough to contain all possible n-grams.

[2] In Chinese, a word can comprises one or multiple characters, which can explore both syntactic and semantic meaning better than a character.

[3] High-order n-gram models need much larger training corpus and higher cost of computation and memory, n usually takes no more than 5 in practice.

likely to be used again in the near future [26, 27]. Finally, global context information of the whole document is exploited in speech recognition using latent semantic analysis (LSA) [28–30] or probabilistic latent semantic analysis (pLSA) and latent Dirichlet allocation (LDA) [31]. All such techniques are still based on word co-occurrence frequencies and, hence, are very far away from a deep understanding of natural language.

Recently, many works focus on using common sense knowledge to enrich the linguistic context in NLP tasks. Stocky et al. [11] used common sense knowledge to generate more semantical words in predictive text entry, Lieberman et al. [9] explored how common sense knowledge can be used to remove the nonsensical hypotheses outputs in speech recognition, Hsu and Chen [7] investigated the usefulness of common sense knowledge for image retrieval, and Cambria et al. exploited common sense knowledge for social media marketing [32] and affective common sense reasoning [33]. However, there is no prior work on the use of common sense knowledge for HCTR. In this paper, we introduce three different methods for integrating common sense knowledge in HCTR.

System Overview

The baseline handwritten recognition system without common sense knowledge is introduced in our previous work [18]. To integrate common sense knowledge effectively, we hereby use a two-pass recognition strategy.

Figure 2 shows the block diagram of our system, where seven steps are taken in the first pass recognition (only the last three steps are needed in the second pass recognition), namely

1. each text line is extracted from the input document image;
2. the line image is over-segmented into a sequence of primitive segments (Fig. 3a), and a character may comprise one segment or multiple segments;
3. several consecutive segments are combined to generate candidate character patterns (Fig. 3b), wherein some are valid character patterns, while some are invalid (also called noncharacter);
4. each candidate pattern is classified into several candidate character classes, forming a character candidate lattice (Fig. 3c);
5. each sequence of candidate characters is matched with a lexicon to segment into candidate words, forming a word candidate lattice (Fig. 3d);
6. each word sequence **C** paired with candidate pattern sequence **X** (the pair is called a candidate segmentation-recognition path) is evaluated by multiple contexts, and the optimal path is searched to output the segmentation and recognition result;
7. all text lines results are concatenated to give the document result, which is used for integrating common sense knowledge in the second pass recognition or output.

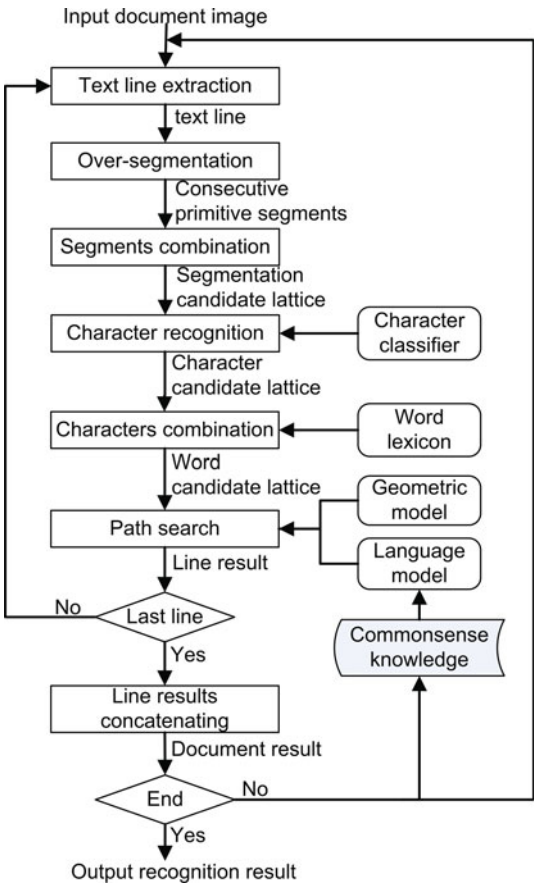

Fig. 2 System diagram for HCTR

In this work, we evaluate each path in the word candidate lattice by integrating character recognition score, geometric context, and linguistic context [18]:

$$f(X^s, C) = \sum_{i=1}^{m} (d_i \cdot lp_i^0 + \sum_{j=1}^{4} \lambda_j \cdot lp_i^j) + \lambda_5 \cdot \log P(C), \quad (1)$$

where m is the character number of candidate word sequence $C = <w_1 \cdots w_l>$ (l is the word number), d_i is the width of the ith character pattern after normalizing by the estimated height of the text line, lp_i^0 is the character recognition score given by a character classifier followed by confidence transformation, $lp_i^j, j = 1, \ldots, 4$ represents the geometric context score given by four geometric models, and log $P(C)$ denotes the linguistic context score usually given by a language model, which will be introduced in details next. The combining weights $\lambda_j, j = 1, \ldots, 5$ are optimized by Maximum Character Accuracy training [18].

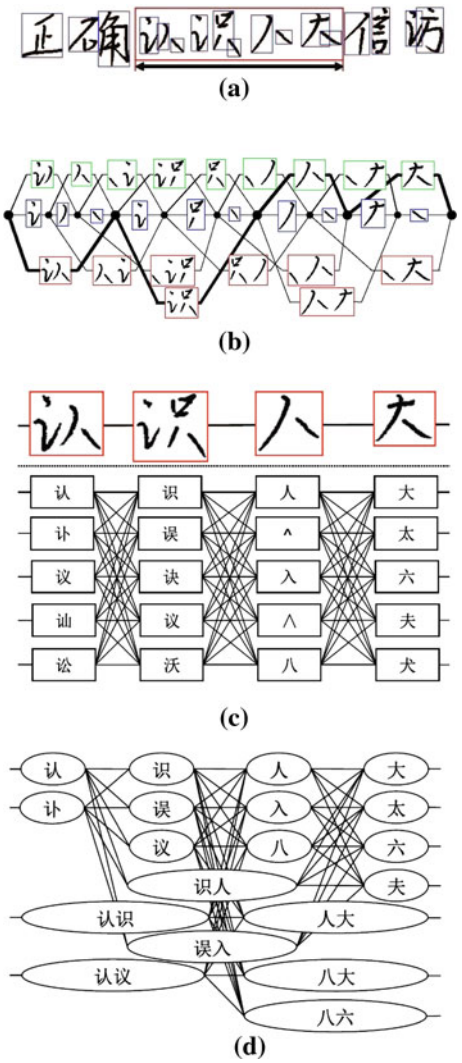

Fig. 3 **a** Over-segmentation; **b** Segmentation candidate lattice; **c** Character candidate lattice of a segmentation (thick path) in (**b**); **d** Word candidate lattice of (**c**)

Under this path evaluation function, we use a refined beam search method [18] to find the optimal path. The search proceeds in frame-synchronous fashion with two steps pruning: first, we only retain the best partial path at each candidate character and then retain the top beam-width partial paths from all the retained paths after the first pruning step ended at each segmented point (many candidate characters with the best partial paths in the first pruning step end at the same segmented point due to the over-segmentation and character classification, see Fig. 3b, c).

Since words can explore the semantic meaning associated with text better than characters in Chinese, common sense knowledge is usually organized as word level. In this work, we only consider two word-level bi-gram language models from our previous work [18] as the baseline language models and then modify them using the common sense knowledge introduced in next section. The two baseline bi-grams are word bi-gram (wbi) and interpolating word and class bi-gram (iwc):

$$\log P_{\text{wbi}}(C) = \sum_{i=1}^{l} \log p(w_i|w_{i-1}), \tag{2}$$

$$\log P_{\text{iwc}}(C) = \log P_{\text{wbi}}(C) + \lambda_6 \cdot \log P_{\text{wcb}}(C), \tag{3}$$

where $\log P_{\text{wcb}}(C)$ is the word class bi-gram (wcb)

$$\log P_{\text{wcb}}(C) = \sum_{i=1}^{l} \log p(w_i|W_i)p(W_i|W_{i-1}), \tag{4}$$

and the term W_i is the class of word w_i using the word clustering algorithm in [34].

Common Sense Knowledge

Common sense knowledge represents human general knowledge acquired from the world and can aid machines to better interpret natural language and select relevant results that make sense in context, rather than simply basing the analysis on word co-occurrence frequencies in the standard n-gram language model. Moreover, because of the data sparsity (according to the Zipf's law [35]), the maximum likelihood estimation (MLE) of n-gram probability is usually improper, especially for those unseen n-grams in the corpus of training texts (no corpus can contain all possible n-grams). Finally, the number of parameters in high-order model (large n) is intractable. In practice, bi-gram ($n = 2$) is usually used, which considers very short-distance context (only one history word).

In this section, we first give a brief introduction of ConceptNet, which is used as a common sense knowledge base in this work; next, we describe how we embed common sense knowledge in the n-gram model to make the probability estimation more reliable; then, we introduce the common sense knowledge as a direct model to interpolate with n-grams for the long linguistic context; finally, we explore the combination of such two models.

ConceptNet

In this work, we select ConceptNet [1, 3] as common sense knowledge. ConceptNet is collected from the Open Mind Common Sense (OMCS) project since 2000 by asking volunteers on the Internet to enter common sense knowledge into the system.

It is a semantic network of concepts connected by relations such as 'IsA,' 'HasProperty,' or 'UsedFor,' which provide a connection between natural language text and an

understanding of the world [36], and has been widely used in many NLP tasks [9, 11, 12]. Figure 4 shows an example of a small section of ConceptNet.

To identify the similarities between concepts in ConceptNet, the AnalogySpace [37] process is employed by applying truncated singular value decomposition (SVD) on the matrix representation of the common sense knowledge base, whose rows are concepts, and columns are the features as the relationships with other concepts. In the AnalogySpace, each concept is represented as a vector, which can be used for further analysis, for example, get the similarity between two concepts.

Embedding Model

To overcome the problem of MLE for unseen n-grams (zero probability), smoothing techniques [38] are usually used, such as Katz back-off smoothing [39]:

$$p_n(w_2|w_1) = \begin{cases} p_{ml}(w_2|w_1) & \text{if } C(w_1, w_2) > 0 \\ \alpha(w_1) \cdot p_n(w_2) & \text{if } C(w_1, w_2) = 0 \end{cases}, \quad (5)$$

where $C(.)$ denotes the n-gram counts in the training corpus, $p_{ml}(.)$ is the direct MLE probability, and $\alpha(w_1)$ is the scaling factor to make sure the smoothed probabilities normalized to one. Although such smoothing can avoid the zero probability problem, it is still based on the MLE of the n-gram itself without considering the relationship between similar words.

Based on the fact that the similar words are usually with similar n-gram properties (e.g., the similar linguistic context), it is very likely that some words are observed in some n-grams in the training text corpus, while their similar words are unseen because of the limited corpus size. Utilizing such similarity, we can get more reliable probability estimation for these unseen n-grams by embedding the word similarity with normal n-gram probability as follows:

$$p_e(w_2|w_1) = \sum_{k=1}^{K} p_n(w_2|w_k^1) p_s(w_k^1|w_1), \quad (6)$$

where the number K denotes the size of similar word set used in our system. We only consider the top K similar words while viewing the similarities of the remaining words as zero, in order to reduce the computation cost due to the large size of Chinese word lexicon (usually more than 0.1 million). The word w_k^1 represents the kth similar word of w_1, $p_n(w_2|w_k^1)$ is the normal bi-gram probability by (5), and $p_s(w_k^1|w_1)$ is the word similarity probability using the common sense knowledge.

Each word in the common sense knowledge database can be represented as a semantic vector using truncated SVD on the common knowledge matrix, as in the AnalogySpace process [37]. With such semantic vectors, we can calculate the word similarity via the cosine function:

$$\text{sim}(w_1, w_k^1) = \cos(\mathbf{x_1}, \mathbf{x_k^1}) = \frac{\mathbf{x_1}^{\mathrm{T}} \cdot \mathbf{x_k^1}}{\|\mathbf{x_1}\| \cdot \|\mathbf{x_k^1}\|}, \quad (7)$$

where $\mathbf{x}$ represents a semantic vector, and $\|\mathbf{x}\|$ denotes the Euclidean norm of the vector. This cosine similarity is widely used in LSA language models [28–30] and other NLP tasks [35].

To get the similarity probability formation in (6), we normalize cosine similarity (7) by all words in the similar word set of word w_1 according to the method in [28]. First, we get the smallest similarity of word w_1:

$$\text{MinSim}(w_1) = \min_{k=1}^{K} \text{sim}(w_1, w_k^1). \quad (8)$$

Next, subtracting out the $\text{MinSim}(w_1)$ from all cosine similarities of word w_1 to assure all similarities larger than zero, then normalizing to get a first probability:

$$p_s'(w_k^1|w_1) = \frac{\text{sim}(w_1, w_k^1) - \text{MinSim}(\text{w}_1)}{\sum_{i=1}^{K} [\text{sim}(w_1, w_i^1) - \text{MinSim}(w_1)]}. \quad (9)$$

Finally, the similarity probability is estimated by a power transformation of the above probability and renormalizing:

$$p_s(w_k^1|w_1) = \frac{p_s'(w_k^1|w_1)^{\gamma}}{\sum_{i=1}^{K} p_s'(w_i^1|w_1)^{\gamma}}, \quad (10)$$

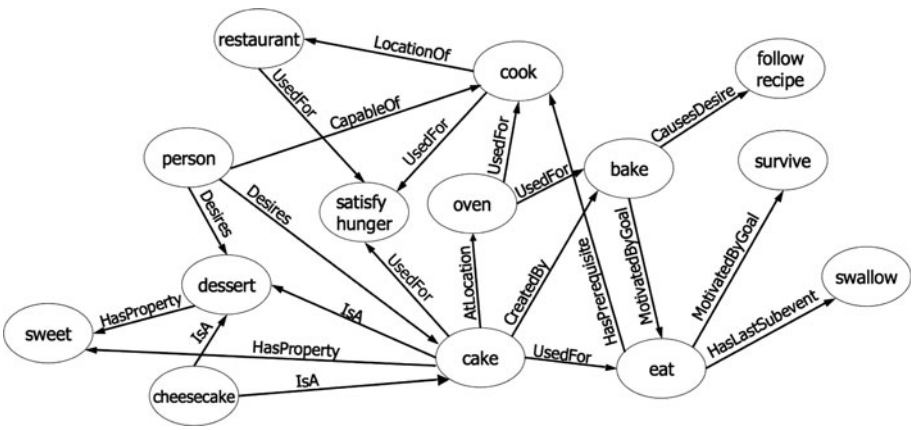

Fig. 4 A small section of ConceptNet [37]

where the power constant γ is set empirically on a data set of validation samples. In the experiments, we can see that the recognition performance is improved by introducing γ.

Finally, it is easy to extent our embedding probability estimation (6) to high-order n-gram model by replacing the normal bi-gram probability $p_n(w_2|w_k^1)$. In this work, we call such approach embedding model since it embeds the common sense knowledge in normal probability estimation of n-gram model.

Direct Model

To make up for the short-distance context limit of the n-gram model (only $n-1$ history words are considered as linguistic context), we calculate the semantic similarity between each word and its all history words using common sense knowledge. A similar approach is commonly adopted in speech recognition, except that LSA is used for semantic similarity [28–30].

In this work, we call this long-distance semantic linguistic context as semantic language model $P_s(C)$, which can be calculated as follows:

$$\log P_s(C) = \sum_{i=1}^{l} \log p_s(w_i|h_i), \tag{11}$$

where l is the number of words in word sequence C, and $h_i = w_1^{i-1} = <w_1 \cdots w_{i-1}>$ $(i > 1)$ is referred as the history of the word w_i ($h_1 = null$, and $p_s(w_1|h_1) = 1$ is assumed). The semantic probability $p_s(w_i|h_i)$ can be further decomposed into the product of word-dependent terms:

$$\log p_s(w_i|h_i) = \frac{1}{i-1} \sum_{j=1}^{i-1} f(w_i, w_j), \tag{12}$$

where $\frac{1}{i-1}$ is a normalization factor to overcome the effect of sequence length of h_i $(i > 1)$, and $f(w_i, w_j)$ is the logarithm of semantic similarity probability of each two words:

$$f(w_i, w_j) = \log \frac{1 + \cos(\mathbf{x_i}, \mathbf{x_j})}{2}, \tag{13}$$

where $\mathbf{x_i}$ and $\mathbf{x_j}$ are the semantic vector of word w_i and w_j, respectively, and the cosine function is defined as (7).

In the above, the semantic language model $P_s(C)$ only considers the history words as the linguistic context h_i in (11). Sometimes, the current word w_i is also influenced by its subsequent words. To consider the whole linguistic context of the document, we use two-pass recognition strategy [24]. In the first pass, we use the baseline system without the semantic language model to get a first recognition result and then use this result to get the semantic language model $P_s(C)$ used in the second pass recognition:

$$\log P_s(C) = \sum_{i=1}^{l} \log p_s(w_i|\text{doc}), \tag{14}$$

where doc is the first pass recognition result, and the logarithm of whole semantic probability $\log p_s(w_i|\text{doc})$ is calculated similarly as (12):

$$\log p_s(w_i|\text{doc}) = \frac{1}{d} \sum_{j=1}^{d} f(w_i, w_j), \tag{15}$$

where d is the number of words in the first result doc. Compared to (11), the whole document semantic language model (14) also considers the linguistic context of subsequent words, although there are some error words in the first pass recognition.

No matter the semantic language model considering only history (11) or whole document (14), the semantic linguistic context is usually not enough to predict the next word. We combine it with normal n-gram model as follows [28]:

$$\log P(C) = \log P_n(C) + \lambda_7 \cdot \log P_s(C), \tag{16}$$

where $P_n(C)$ is a normal n-gram model (e.g., word bi-gram (2) or interpolating word and class bi-gram (3)), and $P_s(C)$ is the semantic language model by (11) or (14). The logarithm is used for more general purposes in path evaluation function (1), and the weight λ_7 is used to balance such two models, which is optimized together with other combining weights in (1) by Maximum Character Accuracy training [18].

We call this approach direct model of integrating common sense knowledge, since we use the common sense knowledge directly as semantic language model $P_s(C)$ in (16).

Moreover, we can get an ensemble model by combining direct model (16) and embedding model (6). Obviously, the normal n-gram model $P_n(C)$ in (16) can be computed by the embedding n-gram probability $p_e(w_2|w_1)$ of (6) instead of the normal n-gram probability $p_n(w_2|w_1)$, and then the ensemble model can be formulated by

$$\log P(C) = \log P_e(C) + \lambda_7 \cdot \log P_s(C), \tag{17}$$

where $P_e(C)$ and $P_s(C)$ are estimated by the embedding model and direct model, respectively.

Experiments

We use the system introduced detailedly in [18] as the baseline (except candidate character augmentation and using language model compression [24]) to evaluate the effect of common sense knowledge in HCTR, and all the experiments are implemented on a desktop computer of

2.66 GHz CPU, programming using Microsoft Visual C++.

Database and Experimental Setting

We evaluate the performance on a large database CASIA-HWDB [22], which is divided into a training set of 816 writers and a test set of other 204 writers. The training set contains 3,118,477 isolated character samples of 7,356 classes and 4,076 pages of handwritten texts (including 1,080,017 character samples). We tested on the unconstrained texts including 1,015 pages, which were segmented into 10,449 text lines and there are 268,629 characters.

The character classifier used in our system is a modified quadratic discriminant function (MQDF) [40], and the parameters were learned from 4/5 samples of training set, and the remaining 1/5 samples were for confidence parameter estimation. For parameter estimation of the geometric models, we extracted geometric features from 41,781 text lines of training text pages. The normal word-level bi-grams were trained on a large corpus (containing about 32 million words and the size of word lexicon is about 0.28 million) from the Chinese Linguistic Data Consortium (CLDC).

We extracted 14,199 English concepts from ConceptNet, which are translated into Chinese using machine translation techniques, and only the concepts in the normal word lexicon from CLDC corpus are retained. A very small number of concepts are used in our experiments, that is, 7,844 concepts (only 2.8 % of the word lexicon). On obtaining the context models, the combining weights in path evaluation function (1) were learned on 300 pages of training text. In addition, another 200 pages of training text are used as validation samples to set the size (K) of similar word set in (6) and the power constant (γ) in (10) by trail and error.

We evaluate the recognition performance using two character-level accuracy metrics as in the baseline system [18]: correct rate (CR) and accurate rate (AR):

$$\begin{aligned} \mathrm{CR} &= (N_t - D_e - S_e)/N_t, \\ \mathrm{AR} &= (N_t - D_e - S_e - I_e)/N_t, \end{aligned} \tag{18}$$

where N_t is the total number of characters in the transcript. The numbers of substitution errors (S_e), deletion errors (D_e), and insertion errors (I_e) are calculated by the aligning the recognition result string with the transcript by dynamic programming. It is suggested that the AR is an appropriate measure for document transcription, while CR is a good metric for tasks of content modeling.

Experimental Results

We evaluate the effect of the common sense knowledge as embedding model, direct model, and combination model. In the embedding model, the size of similar word set is set to 10 (this is, $K = 10$ in (6), it is chosen by trial and error on our validation data set). We also give the processing time on all test pages (1,015 pages) excluding that of over-segmentation and character recognition, which are stored in advance.

First, we evaluated the effect of power transformation in embedding model of integrating common sense knowledge in HCTR (power constant $\gamma = 7$ is chosen by trial and error on the validation data set), and the results of two word-level bi-grams (wbi and iwc) are shown in Table 1. Compared to the embedding model without power transformation [i.e., $\gamma = 1$ in (10)], we can see that the performance of both CR and AR is improved after power transformation, especially in the word bi-gram model. The benefit of power transformation is attributed to the fact that it enlarges the difference of the similarity probabilities in the similar word set of w_1 and helps those with higher similarity. We hence used the power transformation in the experiments of the ensemble model next.

Table 2 shows the results of integrating common sense knowledge using direct model with the context of both history and whole document. Comparing such two direct

Table 1 Effects of power transformation in embedding model

	wbi		iwc	
Models	CR (%)	AR (%)	CR (%)	AR (%)
No power	90.88	90.23	91.19	90.57
Power	91.00	90.35	91.22	90.58

Table 2 Results of direct model with the context of history and whole document

	wbi			iwc		
Models	CR (%)	AR (%)	Time (h)	CR (%)	AR (%)	Time (h)
History	90.99	90.34	1.03	91.22	90.58	1.18
Document	91.00	90.35	2.58	91.23	90.60	2.88

Table 3 Results of common sense knowledge in HCTR

	wbi			iwc		
Models	CR (%)	AR (%)	Time (h)	CR (%)	AR (%)	Time (h)
Baseline	90.98	90.33	0.81	91.21	90.57	0.96
Embedding	91.00	90.35	1.04	91.22	90.58	1.18
Direct	91.00	90.35	2.58	91.23	90.60	2.88
Ensemble	91.00	90.35	2.59	91.23	90.59	2.89

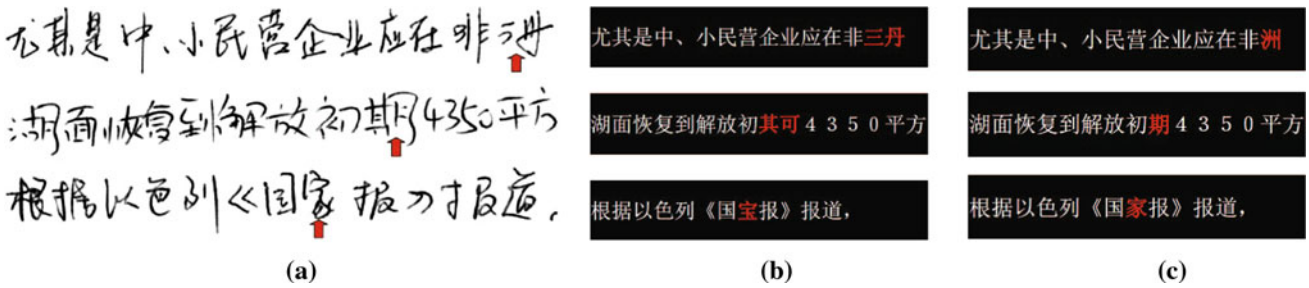

Fig. 5 **a** Three text line images; **b** Recognition results without common sense knowledge; **c** Recognition results with common sense knowledge

models, we can see the performance of model with whole document context is a little higher, because the span of linguistic context is much larger than the history words, although there are some errors in the first pass recognition result. On the other hand, computational time is higher because of two-pass recognition and more calculations of words' semantic similarities in (15).

Finally, we investigate the effect of integrating common sense knowledge in HCTR by three models (embedding, direct, and the ensemble model). In this experiment, the power transformation is used in the embedding model, and the direct model is with the context of whole document from the first pass recognition. All the results are shown in Table 3. Compared to the baseline system without any common sense knowledge, both embedding and direct model, either with wbi or iwc, improve the performances of both CR and AR, and the direct model is a little better; this justifies the importance of long-distance linguistic context, while the ensemble model does not improve the performance further. On the other hand, the processing time is much more in the direct and ensemble model.

Figure 5 shows three recognition examples using direct model of common sense knowledge. We can see that the common sense knowledge does benefit the HCTR system. Three errors (Fig. 5b) made by baseline system without common sense knowledge. Obviously, these errors are nonsensical according to the common sense knowledge, and all are corrected by the direct model (Fig. 5c).

Discussion

Our experimental results demonstrate that the proposed approach of integrating common sense knowledge improves handwriting text recognition performance. Such an improvement, however, is not substantial as the list of common sense knowledge concepts adopted in our experiments is very small: most of the candidate words (about 92 %) in the lattice, in fact, are not contained in the concept list.

For the embedding model, the probabilities of such unseen words are computed back to the normal bi-gram (i.e., $p_e(w_2|w_1) = p_n(w_2|w_1)$ for the word w_1 out of the concept list), and the semantic similarity probabilities in the direct model are set to an empirical value (this is, $f(w_i,w_j) = \log 0.5$) for the unseen words either w_i or w_j. In addition, there are a few errors in the machine translation from English concepts to Chinese. Finally, only the valid words in the concept list contribute to improve the performance (only 41 % words of ground truth text are contained in this concept list at the present time). To get good performances, the concept list needs to be replenished and improved further. In addition, our approach can be also used in other context, such as speech recognition or handwriting recognition of other language text.

Conclusion

This paper presented a novel approach to handwritten Chinese text recognition based on three common sense knowledge-based methods: the embedding model, the direct model, and the ensemble model. The embedding model uses the semantic similarity to make the n-gram probabilities estimation more reliable, especially for those unseen n-grams in the training text corpus. The direct model considers the linguistic context of the whole document to make up for the short-distance context limit of normal n-gram language model. The ensemble model is the combination of such two models. Experimental results showed that common sense knowledge yields improvement in recognition performance by selecting the results that make sense in context, despite the reduced concept list hereby employed. In the future, we plan to expand the Chinese common sense knowledge employed and to develop novel techniques for concept matching.

Acknowledgments This work has been supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316302, the National Natural Science Foundation of China (NSFC) Grants 60825301 and 60933010, and the Royal Society of Edinburgh (UK) and the Chinese Academy of Sciences within the China-Scotland SIPRA (Signal Image Processing Research Academy) Programme. The authors would like to thank Jia-jun Zhang for his aid in the machine translation process.

References

1. Liu H, Singh P. ConceptNet—a practical commonsense reasoning tool-kit. BT Tech J (2004); 22(4):211–26.
2. Lenat D, Guha R. Building large knowledge-based systems: representation and inference in the Cyc project. Reading: Addison-Wesley; (1989).
3. Speer R, Havasi C. Representing general relational knowledge in ConceptNet 5. In: International conference on language resources and evaluation (LREC); 2012. p. 3679–86.
4. Cambria E, Havasi C, Hussain A. SenticNet 2: a semantic and affective resource for opinion mining and sentiment analysis. In: Florida artificial intelligence research society conference (FLAIRS); 2012. p. 202–7.
5. Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In: SIGMOD Conference; 2008. pp. 1247–50.
6. Cambria E, Song Y, Wang H, Hussain A. Isanette: a common and common sense knowledge base for opinon mining. In: International conference on data mining; 2011. p. 315–22.
7. Hsu MH, Chen HH. Information retrieval with commonsense knowledge. In: ACM SIGIR conference on research and development in information retrieval; 2006. p. 651–2.
8. Cambria E, Hussain A. Sentic computing: techniques, tools, and applications. Berlin: SpringerBriefs in Cognitive Computation, Springer (2012).
9. Lieberman H, Faaborg A, Daher W, Espinosa J. How to wreck a nice beach you sing calm incense. In: International conference on intelligent user interfaces; 2005.
10. Cambria E, Hussain A. Sentic Album: content-, concept-, and context-based online personal photo management system. In: Cognitive Computation, Springer, Berlin/Heidelberg; 2012 (in press).
11. Stocky T, Faaborg A, Lieberman H. A commonsense approach to predictive text entry. In: International conference on human factors in computing systems; 2004.
12. Cambria E, Benson T, Eckl C, Hussain A. Sentic PROMs: application of sentic computing to the development of a novel unified framework for measuring health-care quality. Expert Syst Appl (2012); 39(12):10533–43.
13. Dai RW, Liu CL, Xiao BH. Chinese character recognition: history, status and prospects. Front Comput Sci China (2007); 1(2):126–36.
14. Tang HS, Augustin E, Suen CY, Baret O, Cheriet M. Spiral recognition methodology and its application for recognition of Chinese bank checks. In: International workshop on frontiers in handwriting recognition (IWFHR); 2004. p. 263–8.
15. Wang CH, Hotta Y, Suwa M, Naoi S. Handwritten Chinese address recognition. In: International workshop on frontiers in handwriting recognition (IWFHR); 2004. p. 539–44.
16. Su TH, Zhang TW, Guan DJ, Huang HJ. Off-line recognition of realistic chinese handwriting using segmentation-free strategy. Pattern Recogn (2009); 42(1):167–82.
17. Li NX, Jin LW. A Bayesian-based probabilistic model for unconstrained handwritten offline Chinese text line recognition. In: International conference on systems man and cybernetics (SMC), 2010. p. 3664–68.
18. Wang QF, Yin F, Liu CL. Handwritten Chinese text recognition by integrating multiple contexts. IEEE Trans Pattern Anal Mach Intell (2012); 34(8):1469–81.
19. Li YX, Tan CL, Ding XQ. A hybrid post-processing system for offline handwritten Chinese script recognition. pattern analysis and applications (2005); 8(3):272–86.
20. Rosenfeld R. Two decades of statistical language modeling: where do we go from here? Proc IEEE (2000); 88(8):1270–8.
21. Goodman JT. A bit of progress in language modeling: extended version. Technical Report MSR-TR-2001-72, Microsoft Research 2001.
22. Liu CL, Yin F, Wang DH, Wang QF. CASIA online and offline Chinese handwriting databases. In: International conference on document analysis and recognition (ICDAR); 2011. p. 37–41.
23. Wang QF, Yin F, Liu CL. Integrating language model in handwritten Chinese text recognition. In: International conference on document analysis and recognition (ICDAR); 2009. p. 1036–40.
24. Wang QF, Yin F, Liu CL. Improving handwritten Chinese text recognition by unsupervised language model adaptation. In: International workshop on document analysis systems (DAS); 2012. p. 110–4.
25. Siu M, Mari O. Variable n-grams and extensions for conversational speech language modeling. IEEE Trans Speech Audio Process (2000); 8:63–75.
26. Kuhn R, De Mori R. A cache-based natural language model for speech reproduction. IEEE Trans Pattern Anal Mach Intell (1990); 12(6):570–83.
27. Kuhn R, De Mori R. Correction to a cache-based natural language model for speech reproduction. IEEE Trans Pattern Anal Mach Intell (1992); 14(6):691–2.
28. Coccaro N, Jurafsky D. Towards better integration of semantic predictors in statistical language modeling. In: International conference on spoken language processing (ICSLP), 1998. p. 2403–6.
29. Bellegarda JR. A multispan language modeling framework for large vocabulary speech recognition. IEEE Trans Speech Audio Process (1998); 6(5):456–467.
30. Bellegarda JR. Exploiting latent semantic information in statistical language modeling. Proc IEEE (2000); 88(8):1279–96.
31. Mrva D, Woodland PC. Unsupervised Language Model Adaptation for Mandarin Broadcast Conversation Transcription. In: Interspeech, 2006. pp. 2206–9.
32. Cambria E, Grassi M, Hussain A, Havasi C. Sentic computing for social media marketing. Multimed Tools Appl (2012); 59(2): 557–77.
33. Cambria E, Olsher D, Kwok K. Sentic activation: a two-level affective common sense reasoning framework. In: Association for the Advancement of Artificial Intelligence (AAAI); 2012.
34. Martin S, Liermann J, Ney H. Algorithms for bigram and trigram word clustering. Speech Commun (1998); 24(1):19–37.
35. Manning CD, Schutze H. Foundations of statistical natural language processing. 2nd ed. The MIT Press, Cambridge (1999).
36. Lieberman H, Liu H, Singh H, Barry B. Beating common sense into interactive applications. AI Mag (2004); 25(4):63–76.
37. Speer R, Havasi C, Lieberman H. AnalogySpace: reducing the dimensionality of common sense knowledge. In: Association for the Advancement of Artificial Intelligence (AAAI). 2008. p. 548–53.
38. Chen SF, Goodman J (1998) An empirical study of smoothing techniques for language modeling. Cambridge, Massachusetts: Computer Science Group, Harvard University, TR-10-98.
39. Katz SM. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans Acoustics Speech Signal Process (1987); 35(3):400–1.
40. Kimura F, Takashina K, Tsuruoka S, Miyake Y. modified quadratic discriminant functions and the application to Chinese character recognition. IEEE Trans Pattern Anal Mach Intell (1987); 9(1):149–53.