IJDAR
DOI 10.1007/s10032-015-0251-1

ORIGINAL PAPER

Handwritten word recognition using Web resources and recurrent neural networks

Cristina Oprean[1] · **Laurence Likforman-Sulem**[1] · **Adrian Popescu**[2] · **Chafic Mokbel**[3]

Received: 10 April 2014 / Revised: 23 July 2015 / Accepted: 24 July 2015

Abstract Handwriting recognition systems usually rely on static dictionaries and language models. Full coverage of these dictionaries is generally not achieved when dealing with unrestricted document corpora due to the presence of Out-Of-Vocabulary (OOV) words. We propose an approach which uses the World Wide Web as a corpus to improve dictionary coverage. We exploit the very large and freely available Wikipedia corpus in order to obtain dynamic dictionaries on the fly. We rely on recurrent neural network (RNN) recognizers, with and without linguistic resources, to detect words that are non-reliably recognized within a word sequence. Such words are labeled as non-anchor words (NAWs) and include OOVs and In-Vocabulary words recognized with low confidence. To recognize a non-anchor word, a dynamic dictionary is built by selecting words from the Web resource based on their string similarity with the NAW image, and their linguistic relevance in the NAW context. Similarity is evaluated by computing the edit distance between the sequence of characters generated by the RNN recognizer exploited as a filler model, and the Wikipedia words. Linguistic relevance is based on an N-gram language model estimated from the Wikipedia corpus. Experiments conducted on a word-segmented version of the publicly available RIMES database show that the proposed approach can improve recognition accuracy compared to systems based on static dictionaries only. The proposed approach shows even better behavior as the proportion of OOVs increases, in terms of both accuracy and dictionary coverage.

✉ Laurence Likforman-Sulem
likforman@telecom-paristech.fr

[1] Telecom-Paristech, Paris, France

[2] CEA List, Paris, France

[3] University of Balamand, Al Koura, Lebanon

Keywords Handwritten word recognition · Out-Of-Vocabulary word · Web resources · Dynamic dictionary · Recurrent neural networks

1 Introduction

While largely solved in controlled conditions, handwriting recognition is still a field of research in a general case. Recognition systems have to cope with variability of both character shapes and content. Character shapes vary with styles and individuals. Variability can be partly solved by increasing the size of training sets and using robust recognition systems such as hidden Markov models (HMMs)- and recurrent neural networks (RNNs)-based systems. Hidden Markov modeling can cope with nonlinear distortions, while RNNs can learn distant dependencies between observations.

Recognition systems, HMMs or RNNs, also rely on linguistic resources including (static) dictionaries and language models in order to direct the recognition process. Thus, performance of recognition systems is tributary to an appropriate choice of the size of their static dictionary. Large dictionaries ensure good coverage, but accuracy drops due to higher confusion between words that look similar. Conversely, small size dictionaries achieve high accuracy, but fail to cover a large proportion of the words to recognize. Thus, when documents include a limited number of topics, a dictionary limited to these topics ensures better performance. But when considering less focused collections, such as unconstrained mails (see Fig. 1) or historical archives, the vocabulary tends to be much larger, increasing the size of static dictionaries.

Since any limited-size dictionary fails to be exhaustive, there are always words that are not included. These words are called Out-of-Vocabulary (OOV) words and are usually the least frequent words that were removed when creating

Published online: 13 August 2015

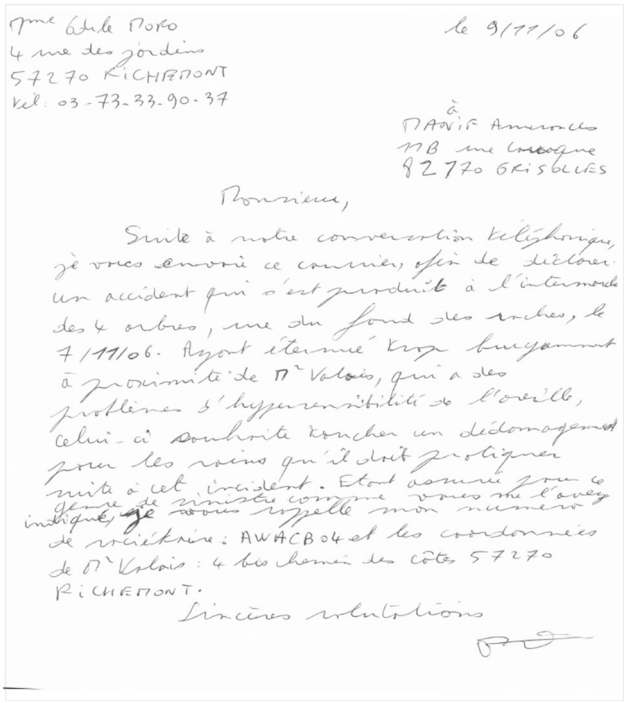


Fig. 1 Sample of unconstrained handwritten mail (RIMES database)

static dictionaries. In addition to the non-frequent words, OOVs include named entities (e.g., first or last names, geographic locations, phone numbers, dates, company names, ages, bank account numbers) that were not encountered in training resources, words associated with new topics which appear over time, words from other languages embedded in texts, grammatical forms of verbs or nouns which were not present in training resources (e.g., "signale" but not "signalés").

OOV recognition can be handled using different approaches. One way to deal with OOVs is to build open-vocabulary systems such as filler models. In such models, any character sequence can be output, however, a character N-gram model guides the recognition in order to improve performance [1,2]. However, when no static dictionary is used, the recognition performance drastically drops. More recently, vocabulary enrichment was achieved by decomposing the lexicon based on a morphological analysis [3]. The new vocabulary is a combination of words and subwords (roots, prefixes and suffixes) obtained as a result of the decomposition. Although theoretically interesting, this method is complex and improves results only by a small margin. OOV recognition is also important in speech recognition. In this domain, recent works exploit Web resources to recover OOVs [4,5]. The local context of a word detected as OOV is used to query the Web with a search engine.

We propose a new word recognition approach that combines the advantages of static dictionaries with the use of external resources. We consider the Web as a corpus and assume that OOV words are likely to appear in this corpus. We rely on Wikipedia, a large and publicly available Web resource which is constantly updated but other Web resources could be used instead. However, words which appear in the external resource are equally valuable. Inspired by works in NLP (natural language processing, see Sect. 2), we propose to build dynamic dictionaries from this Web resource for non-anchor words (NAWs). In this work, non-anchor words designate both the OOVs and the IV (In-Vocabulary) words not reliably recognized. To recognize a non-anchor word, a dynamic dictionary is built by selecting words from the Web resource based on their string similarity with the NAW image and their linguistic relevance in the NAW context. Similarity is evaluated by computing the edit distance between the character string generated by an RNN recognizer and the Wikipedia words. Linguistic relevance is based on an N-gram language model estimated from the Wikipedia corpus. To our knowledge, this is the first study that leverages large-scale Web resources, such as Wikipedia, for large or open-vocabulary handwriting recognition. Experimental results show that the use of dynamic dictionaries improves recognition accuracy compared to the sole use of a static dictionary. We also show that our approach behaves better as the size of the static dictionary decreases and can also work even if starting without any static dictionary.

The paper is organized as follows: Sect. 2 presents the related work concerning dictionary reduction techniques and NLP approaches using Web as an external resource. Section 3 describes the general overview of the proposed approach. Section 4 presents the baseline RNN recognition system. Wikipedia resources at unigram and bigram levels are described in Sect. 5. Based on an anchor/non-anchor (AW/NAW) word classification, described in Sect. 6.1, dynamic dictionaries are built for the words classified as NAW. Dynamic dictionary construction is presented in Sects. 6.2 and 6.3. The experimental results are provided in Sect. 7. Finally, Sect. 8 is dedicated to summarize the major findings and to propose future work.

2 Related works

Handling large lexicons and coping with unknown words are major issues for recognition tasks such as speech and handwriting recognition. Several approaches have been proposed. The first approach consists in starting with a large static dictionary and filtering it to better fit with the domain of interest. Since dealing with large lexicons increases both computational complexity and confusions with similar words, lexicon reduction approaches have been proposed to filter an initial dictionary.

A simple solution to filter out words from a dictionary is to consider word length as a decision criterion. In [6], an

estimation of the minimal and maximal lengths of a word is performed based on the sequence of feature vectors. These values delimit an interval, and only words whose lengths fall within that interval are selected. In [7,8], the word length is estimated by counting the strokes in the area between upper and lower baselines.

Another approach for lexicon reduction is to consider the shape of the word as a filtering criterion. In [9,10], the stroke types for each character are identified (e.g., ascender, descender, medium). A word is considered to be a concatenation of these symbols. In the recognition phase, the sequence of symbols obtained for a word to recognize is compared to the representation of each word from the dictionary and only those words that are similar are kept. Both approaches for dictionary reduction (by using length and shape) are based on features extracted from word images and are therefore not very robust to noise and variations in writing styles.

Other approaches for dictionary reduction consist in automatically identifying the topics of the processed document, and to use the lexicon of the identified topic for further recognition [11,12]. The topics may be identified from recognition outputs, considering top-N hypotheses [13,14]. Reducing the dictionary to related concepts improves recognition. Lexicon reduction approaches do not cope with OOV words, but rather assume that the word to recognize belongs to the initial dictionary. In contrast, our approach extends the initial static lexicon in order to cope with unknown words. However, our approach also includes a selection step in which the best word candidates from the external resource are kept.

The World Wide Web is an unlimited resource that can be successfully used in a variety of NLP tasks. It is constantly updated, and words that do not belong to general dictionaries are frequently included. It has been recently used as linguistic resources in complement or instead of closed and handcrafted corpuses [15]. The Web can be successfully used for building language resources in different fields, including computational linguistics [16,17], statistical machine translation [18,19], speech recognition [4,20] and spelling corrections. For instance, unknown and sparse N-grams can be estimated through Wikipedia corpus [21]. Part-of-Speech (POS) tagging of unknown words can be achieved through Web search by including in the query the known context of the unknown word [22,23]. In [15], the authors propose a method that uses the Web as a source of misspellings, to automatically build error models. More recently in [24], OCR errors are postprocessed using Google suggestions. In [25], the unigrams provided by the WEB-IT corpus are used to rank hypotheses derived from character recognition outputs. From the results obtained for spelling corrections, it is important to notice that 80 % of misspelling errors can be found at edit distance of one. In handwriting recognition systems, the words to recover are at a larger edit distance and therefore the problem is far more complex [26].

The objectives of the works presented above are similar to ours: dealing with unknown words, absent from an initial corpus or dictionary. These works show that large Web repositories are efficient for coping with unknown words, and we propose to use such repository within a handwriting recognizer.

3 Overview of the proposed approach

We propose a handwriting recognition approach which combines high performance, due to reasonably sized static dictionaries, and flexibility provided by external Web resources. Dynamic dictionaries are built from such resources, enlarging the coverage of the initial static dictionary. We process a sequence of word images as shown in Fig. 2. We consider that this sequence includes anchor words and non-anchor words. Anchors words (AWs) are words reliably recognized, while non-anchor words (NAWs) are the remaining words (OOVs words or non-reliable IV words). In order to differentiate between anchor and non-anchor words, word images are input to a RNN recognizer, exploited in two configurations: without a dictionary as a filler model and with a static dictionary. We use a special kind of RNN, namely a Bidirectional Long Short-Term Memory (BLSTM) that takes into account dependencies between distant observations.

For each NAW, a dynamic dictionary is built by selecting words from the Web resource based on their string similarity with the NAW image, and their linguistic relevance in the NAW context. Similarity is evaluated by computing the edit distance between the sequence of characters generated by the RNN recognizer exploited as a filler model, and the Wikipedia words. Linguistic relevance is based on an N-gram language model estimated from the Wikipedia corpus.

The NAW word image is then re-decoded with the dynamic dictionary. Once an NAW image has been recognized, it becomes an anchor word (AW). The process is iterated till there are no more NAWs.

4 BLSTM recognizer

The recognition system is based on the sliding window approach, thus avoiding word segmentation into characters. A sliding window of width $w = 9$ pixels is shifted from left to right on the word image in order to extract a sequence of feature vectors. However, word images are first preprocessed, deskewed and deslanted by the approach described in [27].

Each sliding window is divided into 20 fixed cells, and 37 features are extracted. These statistical, geometrical and directional features are described in [28]. Two consecutive sliding windows have a shift of $\delta = 3$ pixels (see Fig. 3). These parameters were optimized in [29].

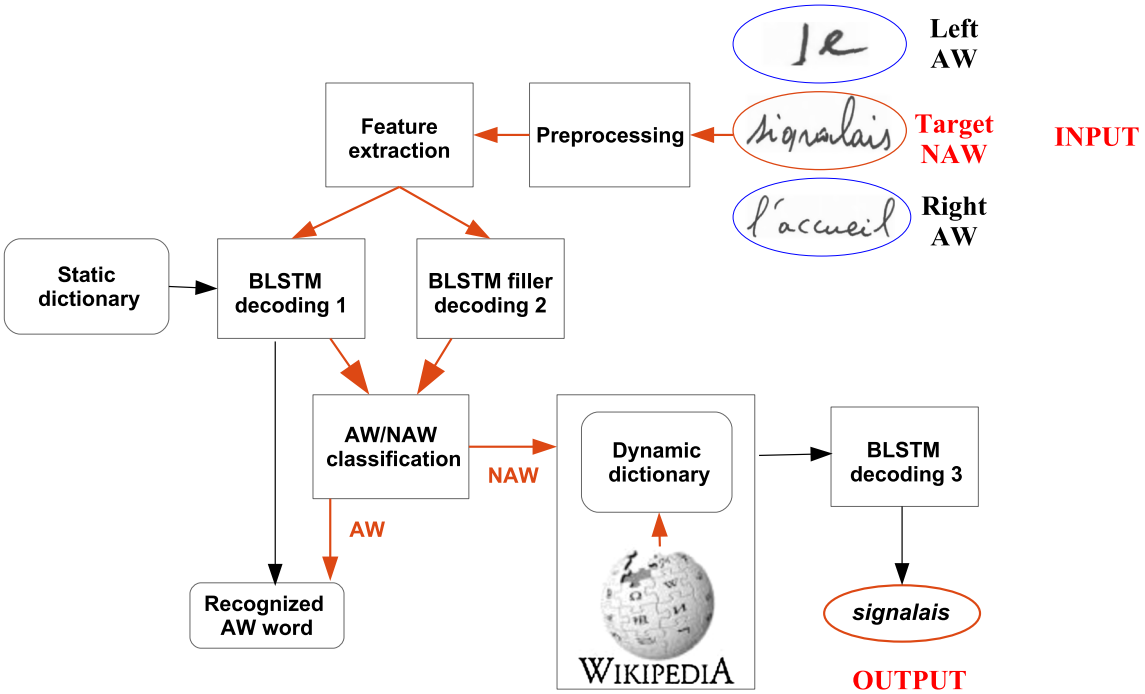


Fig. 2 Word recognition approach with external Wikipedia resources. A dynamic dictionary is built for the non-anchor word (NAW) "signalais", surrounded by two anchor words (AWs), "je" and "l'accueil".

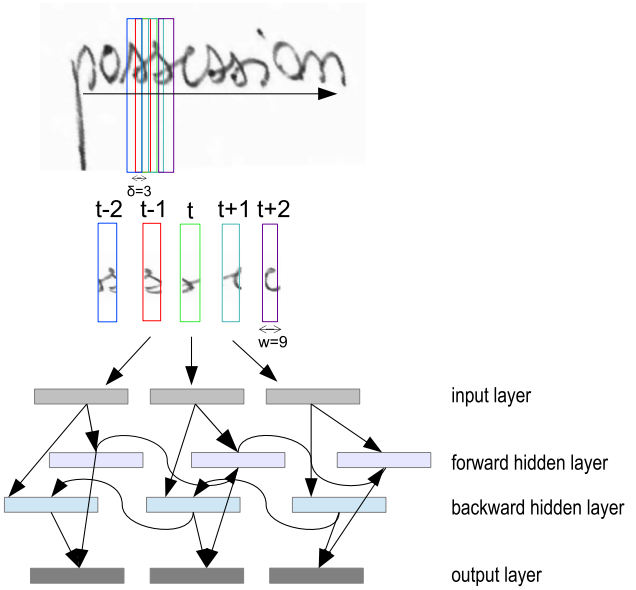


Fig. 3 Bidirectional Long Short-Term Memory (BLSTM) recognizer and sliding windows of size $w = 9$ and shift $\delta = 3$ for the word "possession" from time *t-2* to *t+2*

Recurrent neural networks are a class of artificial neural networks where the connections between hidden units allow dynamic temporal behavior and information storing. Bidirectional RNNs [30] process the data forward and backward by using two separate recurrent layers. Thus, bidirectional RNNs take advantage of the past and future context of the sequence given as input. The forward pass processes the sequence from left to right, while the backward pass takes the input sequence in the opposite direction. Both of them are connected at the same input and output layers (see Fig. 3).

BLSTMs are a type of bi-directional RNN architecture where the summation units in the hidden layer are replaced with memory blocks and they were introduced to solve the vanishing gradient problem [31]. The network input layer is composed of the features extracted from the sliding window at each time t. The output layer at time t includes as much cells as the number of symbols and letters used in the lexicon, e.g., 79 symbols corresponding to all 79 French characters (a–z, A–Z, 0–9, " / ", " ' " , " ", "–" , blank symbol and accentuated characters). Following the work developed in [32], each BLSTM hidden layer has 100 memory blocks. For training the network, the "Back-Propagation Through Time" method [33,34] is used for each utterance. The weights are updated using the gradient descent with momentum method.

For each frame, the posterior probability corresponding to each character class is computed by the BLSTM. These posterior probabilities are given as input to a CTC layer (Connectionist Temporal Classification) [35] which outputs, along with a score, a sequence of characters when no constraint is imposed (case without a dictionary) or a word from a dictionary when a dictionary is used. The CTC implementation is the one introduced in [32,36] which relies on a forward–backward algorithm.

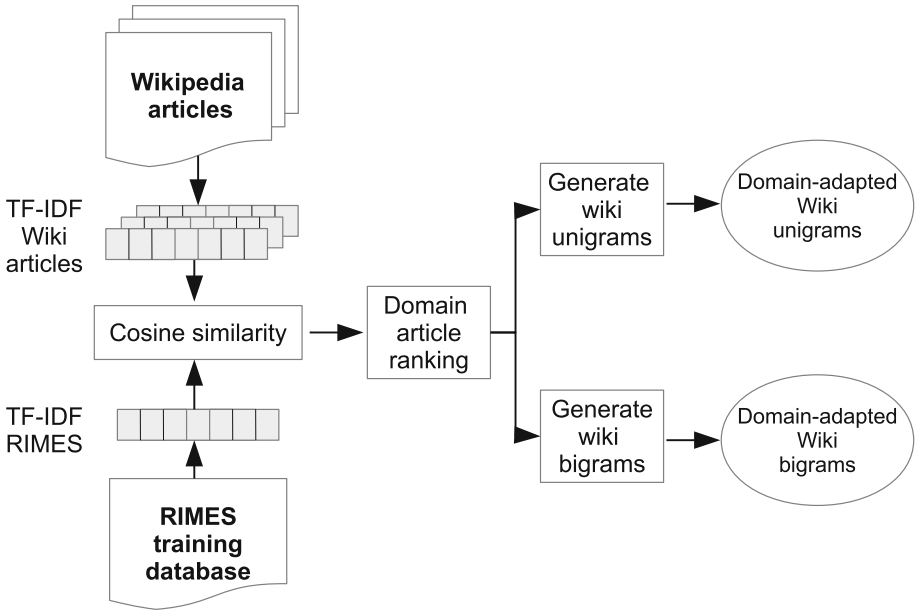


Fig. 4 Generating Wikipedia-based unigrams and bigrams from domain-adapted articles

5 Wikipedia resources

Wikipedia is a comprehensive encyclopedia that describes a large number of concepts and is thus fitted for creating dictionaries or language models with a good language coverage. Compared to other Web corpora, the choice of Wikipedia offers two important advantages. First, the encyclopedia covers a wide range of domains and, therefore, can be effectively used to process handwriting corpora covering a large number of domains. Second, the resource is freely available and constantly updated. A dump of French Wikipedia from September 2012 is used in this work. A total of 410,482 articles that contain each at least 100 distinct words were selected.

Language models vary from one domain to another and their effectiveness is determined by the relatedness between the background collection and the domain representation. Consequently, we first focus on the selection of a Wikipedia subset which is most relevant to the target domain. Wikipedia articles are modeled using a classical TF–IDF (Term Frequency–Inverse Document Frequency) representation [37]. TF–IDF measures how important a word is for a document through TF, but also accounts for its distribution within the Wikipedia collection through IDF. Simply put, the importance of a term for a document is directly proportional with its number of occurrences in the document and inversely related to the number of different documents of a collection in which it appears.

The cosine similarity measure [38] between the TF–IDF representation of the training collection used in experiments (i.e., the RIMES database [39] in our experiments), considered as a single document, and that of each Wikipedia article is then computed. As a result, Wikipedia articles are ranked according to their proximity with the training collection. A domain-adapted dictionary is obtained by parsing the first 20,000 most similar articles, as illustrated in Fig. 4. Only words that occur at least 12 times in these articles are retained. This constraint is useful to discard erroneous words such as typos and non-word strings present in the Wikipedia corpus. In addition, it can be noted that rare Wikipedia words are also rare in the evaluation dataset and, more generally, in written texts. As a consequence, changing the value of this parameter in a range between 1 and 20 results in small performance variation and the best result was obtained with a value of 12 on a validation dataset. The domain-adapted Wikipedia dictionary thus includes around 76,000 unigrams, together with their document frequency (i.e., the number of unique documents in which terms appear).

The 20,000 most similar Wikipedia articles are also used as a corpus for providing word bigrams. For each word in the Wikipedia corpus, we count the number of occurrences of words which appear in the first position to its left and right. The left and right contexts obtained for the word *"raccroche"* are illustrated in Fig. 5. We will refer to the two lists of bigrams as left bigrams and right bigrams, respectively.

6 Exploiting Web-based dynamic dictionaries for word recognition

Dynamic dictionaries are built for non-anchor words. Thus, the proposed approach starts with detecting such non-anchor words by classifying each word in the sequence as anchor word (AW) or non-anchor word (NAW). The classification is described in Sect. 6.1. Words from the external Web resources

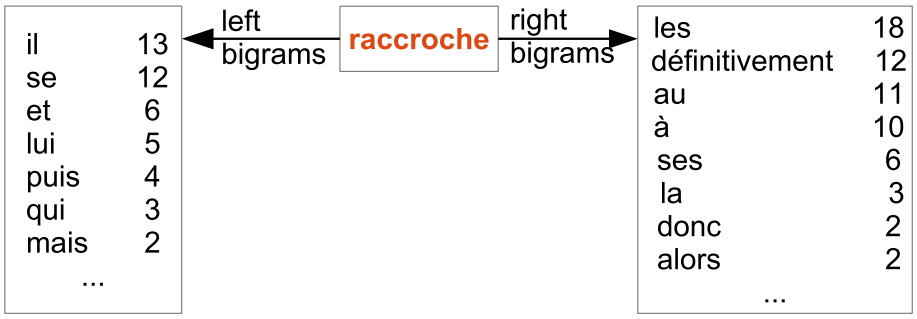


Fig. 5 *Left* and *right* bigrams for word *"raccroche"* extracted from Wikipedia

are selected to build these dictionaries as indicated in Sects. 6.2 and 6.3. The dynamic dictionary built for each NAW is used to re-decode it with the BLSTM recognizer (Sect. 6.4).

6.1 Anchor/non-anchor word classification

Anchor words are reliable In-Vocabulary (IV) words, while non-anchor words are the remaining words, i.e., OOVs and unreliable IVs. Anchor words are thus words from the static dictionary recognized with high confidence. Confidence is measured by the probability provided by the recognizer and by the fact that a vocabulary-independent recognizer (filler model) and a vocabulary-dependent recognizer would yield two similar character strings. Such match/mismatch between phoneme and word-based recognizers has been studied for speech recognition for detecting OOV regions [40–42].

The BLSTM recognizer provides for each word image the best word w from the static dictionary along with its recognition score $L_p(w)$. Similarly, the BLSTM filler provides the vocabulary-independent best character string c associated with w. Both outputs w and c are useful for classifying words as anchors or non-anchors. In order to label a recognized word w as an AW, its recognition score must be greater than a specific threshold and its lexical distance to the corresponding sequence c must be smaller than another threshold.

To compute the distance between a word w and its corresponding character string c, we use the following measure, $distLev$, which is a normalized Levenshtein distance calculated as:

$$distLev(c, w) = \frac{s + d + i}{\max(|c|, |w|)} \tag{1}$$

where s, d and i are the minimum numbers of single character edits such as substitutions, deletions and insertions, respectively, to transform c into w.

Preliminary experiments showed that the optimal thresholds vary with the text to recognize. Therefore, the thresholds must be related to local statistics of the text. We propose to derive local statistics such as the average recognition score $avglogProba$, as well as the average Levenshtein distance $avgdistLev$ between the words and their corresponding character strings, from a subset of words recognized with enough confidence (IV words). This subset is defined by constraining the recognition scores L_p to be greater than a threshold $thre$, found on a validation set.

The local statistics computed from this subset are used within the decision rule which makes the final classification for a given word w, associated with its string c. w is an anchor word if it satisfies the following equations:

$$distLev(c, w) \leq avgdistLev + 0.3 \tag{2}$$

$$L_p(w) \geq avgLogProba + 0.01 \tag{3}$$

Bias values 0.3 and 0.01 are empirically determined on a validation dataset and are used throughout the experiments. The remaining words are non-anchors words (NAWs): They do not satisfy either Eq. (2) or Eq. (3). Dynamic dictionaries are built for NAWs only, while AWs do not need any further processing.

We set the threshold $thre$ on the Rimes word validation set (see Sect. 7.1). We assume that most IV words should belong to the subset of words recognized with enough confidence. Thus, we use the IV/OOV ground truth of the Rimes validation database. Recognition scores for each class, i.e., IV and OOV classes, are collected (see Fig. 6), and $thre$ is set as the average of the recognition scores of the OOV class.

6.2 Exploiting non-anchor word linguistic context

We consider a sequence of n words $w_1, w_2, \ldots, w_n$. The AWs are denoted by $\hat{w}_i$. All the other words are denoted by $\check{w}_j$. It is supposed, without loss of generality, that the number of AWs is m and their indexes belong to the set $I = i_1, \ldots, i_m$. Considering that only the AWs are known, the probability of an NAW $\check{w}_j$, given its context, can be written as:

$$P(\check{w}_j | w_1, \ldots, w_{j-1}, w_j, \ldots, w_n) = P(\check{w}_j | \hat{w}_{i_1}, \ldots, \hat{w}_{i_m}) \tag{4}$$

N-grams can be used to estimate these probabilities. It is well known that the estimation of N-grams requires huge amount

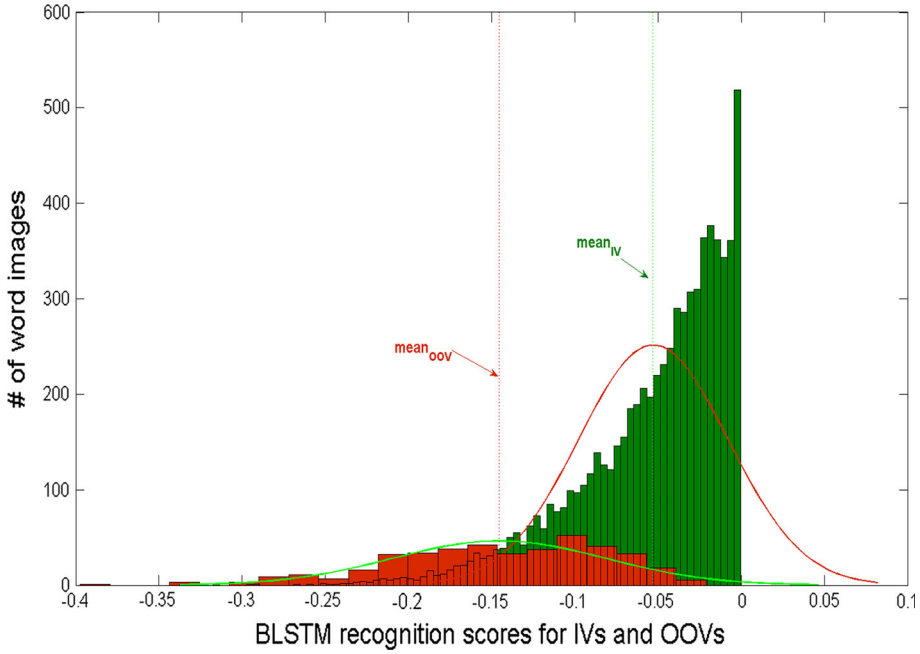


Fig. 6 Score distributions of IV and OOV word images on the validation RIMES dataset. The two distributions are centered on two distinct means mean_{OOV} and mean_{IV}

of data and reliable estimates could only be obtained if the context N is limited. In this case, the conditional probability becomes:

$$P(\breve{w}_j|w_1,\ldots,w_{j-1},w_j,\ldots,w_n) = P(\breve{w}_j|\hat{w}_{i_k},\ldots,\hat{w}_{i_{k+l}}) \quad (5)$$

where $j-N+1 \leq i_k \leq \cdots \leq i_{k+l} \leq j+N-1$.

In the case of $N=2$, bigrams are being used. This approximation introduces an issue related to the fact that the left or right neighbors of a target NAW might not be AWs. In this case, two approaches can be considered:

- An iterative approach, in which at each iteration the dynamic dictionary is constructed for the NAWs which have an AW in their adjacent left and/or right neighborhood and recognize those words based on this dictionary. After one iteration, a part of the NAWs, namely those that have an adjacent AW, is labeled as AWs and the process continues till there is no more NAW.
- Maintain a bigram approximation using probabilities not only estimated on the adjacent neighborhood, but from further ones defined as $P(\breve{w}_j|\hat{w}_{j+k})$ or $P(\breve{w}_j|\hat{w}_{j-k})$, where $k>1$. In this case, the contextual probability can be computed based on the nearest AWs and iterations are no longer needed. However, dynamic dictionaries obtained with large contexts are less robust, because of looser linguistic relations.

In the present work, the iterative approach is adopted. In order to illustrate it, Fig. 7 provides examples of possible

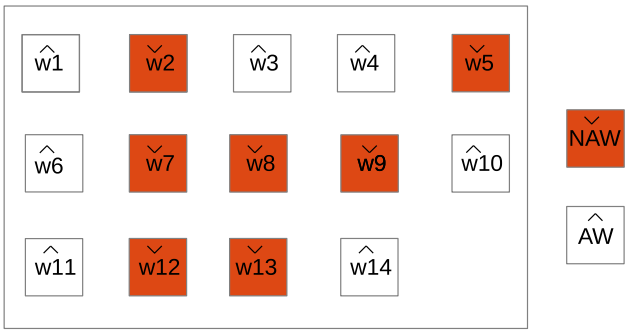


Fig. 7 Example of a document with NAWs (non-anchor words) and AWs (anchor words) in different configurations

scenarios related to the positions of AWs and NAWs. The probability $P(\breve{w}_j|w_1,\ldots,w_n)$ (Eq. 5) has to be computed for each $\breve{w}_j$ from the text. Depending on the configuration of the NAWs and AWs, this probability is estimated differently. The configurations are the following:

- *Case AW–NAW–AW* this is the case for words w_2 and w_5 in Fig. 7, surrounded by w_1, w_3 and w_4, w_6, respectively. For instance, $P(\breve{w}_2|w_1,\ldots,w_{14}) = P(\breve{w}_2|\hat{w}_1,\hat{w}_3,\hat{w}_4,\hat{w}_6,\hat{w}_{10},\hat{w}_{11},\hat{w}_{14})$, is approximated by $P(\breve{w}_2|\hat{w}_1,\hat{w}_3)$. In this case, both left and right contexts of the word w_2 are used.
- *Case AW–NAW–NAW–AW* this case is represented in Fig. 7 on line 3: $\hat{w}_{11}$, $\hat{w}_{12}$, $\hat{w}_{13}$, $\hat{w}_{14}$. The NAWs are constrained only by the left or right context. $P(\breve{w}_{12}|w_1,\ldots,w_{14}) = P(\breve{w}_{12}|\hat{w}_1,\hat{w}_3,\hat{w}_4,\hat{w}_6,\hat{w}_{10},\hat{w}_{11}\hat{w}_{14}) \approx P(\breve{w}_{12}|\hat{w}_{11})$ and $P(\breve{w}_{13}|w_1,\ldots,w_{14}) = P(\breve{w}_{13}|\hat{w}_1,\hat{w}_3,\hat{w}_4,\hat{w}_6,\hat{w}_{10},\hat{w}_{11},\hat{w}_{14}) \approx P(\breve{w}_{13}|\hat{w}_{14})$. Only the left con-

text of the word w_{12} is used, while for the word w_{13} the right context is exploited.

- *Case AW–NAW–NAW–NAW–AW* represents the configuration of an NAW for which a context cannot be built from reliable AW neighbors. As stated earlier, the iterative approach shall be used to cope with this issue. In this example, it works as follows: $\check{w}_8$ is recovered in the iteration following the recovering of $\check{w}_7$ and $\check{w}_9$, which are recovered as in the case *AW–NAW–NAW–AW* using the bigrams $P(\check{w}_7|\hat{w}_6)$ and $P(\check{w}_9|\hat{w}_{10})$. Actually, $\check{w}_7$ and $\check{w}_9$ are labeled AW after their decoding using their dynamic dictionary and the bigrams $P(\check{w}_8|\check{w}_7)$ and $P(\check{w}_8|\check{w}_9)$ can be used to build the dynamic dictionary for $\check{w}_8$.

It is worth noting that for larger contexts *AW–NAW–…–NAW-AW*, several iterations as the latest one may be used considering knowledge from the exterior to the interior.

6.3 Dynamic dictionary construction from Web resource

The best case consists in building the dynamic dictionary based on the bigrams of adjacent AW words. However, it is sometimes not possible to build the dynamic dictionary from bigrams since they may not be available. Two cases illustrate this scenario: (i) The first case corresponds to an application where we start recognition with no static dictionary. Thus there is no AW, inhibiting the possibility of building dynamic dictionaries based on bigrams, (ii) the second case corresponds to NAWs whose contextual words have a too short bigram list, inhibiting the possibility to rely on bigrams only. In such cases, unigrams are used as a backup solution. In the following, the dynamic dictionary creation using unigrams or bigrams is described.

6.3.1 Collecting words from unigrams

Words from the domain-adapted Wikipedia dictionary (see Sect. 5) can be selected to integrate the dynamic dictionary, based on their string similarity to an NAW word. For this selection, the NAW decoded character string and the unigrams of the Wikipedia words are supposed to be available. The Levenshtein distance [43] is used to compare the Wikipedia words to the character string. It computes the number of edits necessary for one sequence to turn into the other: deletions, substitutions and insertions. The most similar Wikipedia words to the entry character string c are grouped based on their Levenshtein distances. At equal distance, Wikipedia words are sorted using their document frequency. The Wikipedia words for which the difference between their length and that of the decoded sequence c is at most l, are retained in the dynamic dictionary while taking care not to have the size of this dictionary exceeding k. Note that the k retained words might also include words selected using bigrams. The values k and l are empirically determined on a validation database and are set to 500 and 5, respectively. Varying the size k of the dynamic dictionary with values ranging between 100 and 1000 has limited influence on global performance. This behavior is determined by the fact that the average Levenshtein distance between NAW sequences obtained with the BLSTM filler and ground truth words is 2.8. With such a mismatch, the ground truth word is often among the nearest neighbors with respect to the Levenshtein distance, and it is not necessary to retain a lot of candidates. For comparison, the authors of [44] report that in spelling correction, 80% of the misspellings have an edit distance of one. These results show that the problem tackled here is more difficult.

The dynamic dictionary creation from unigrams is illustrated in Fig. 8. The word image *"signalais"* is initially decoded as *"sinnxhsas"* when a filler model is used. By computing the Levenshtein distance against all words included in the Wikipedia dictionary, we obtain three groups of words (Levenshtein distances equal to 4, 5 and 6). Note that the ground truth word *"signalais"* is found at a Levenshtein distance equal to five and has a low document frequency.

6.3.2 Collecting words from bigrams

As described above, when bigrams are available in the iterative process, they are used to select the words from the domain-adapted Wikipedia. As previously, the most likely words according to the bigrams are grouped based on their Levenshtein distance to the character string decoded by the filler model. The words to include in the dynamic dictionary are selected first following the increasing Levenshtein distance and second by a decreasing bigram. If the dictionary obtained from neighboring AWs is too rich, only the k most frequent words are retained. If this dictionary is not rich enough, it is complemented with Wikipedia words selected from unigrams (Sect. 6.3.1).

Higher-order N-grams ($N > 2$) could be introduced in the process by including in the dynamic dictionary words collected first from N-grams, then from $N-1$-grams …till unigrams. There should thus be enough AWs after the first recognition step, and accurate N-grams estimates should be available. The preference given here for bigrams is motivated by the fact that we have good bigram estimates which encode accurate linguistic relations between words.

6.4 Word recognition

In order to develop the iterative solution for recovering NAWs, presented in Sect. 6.2, the word sequence is first traversed from top to bottom, collecting for each NAW whose neighbor on its left is an AW, the list of words corresponding to the right bigrams of this AW. Then, we traverse the

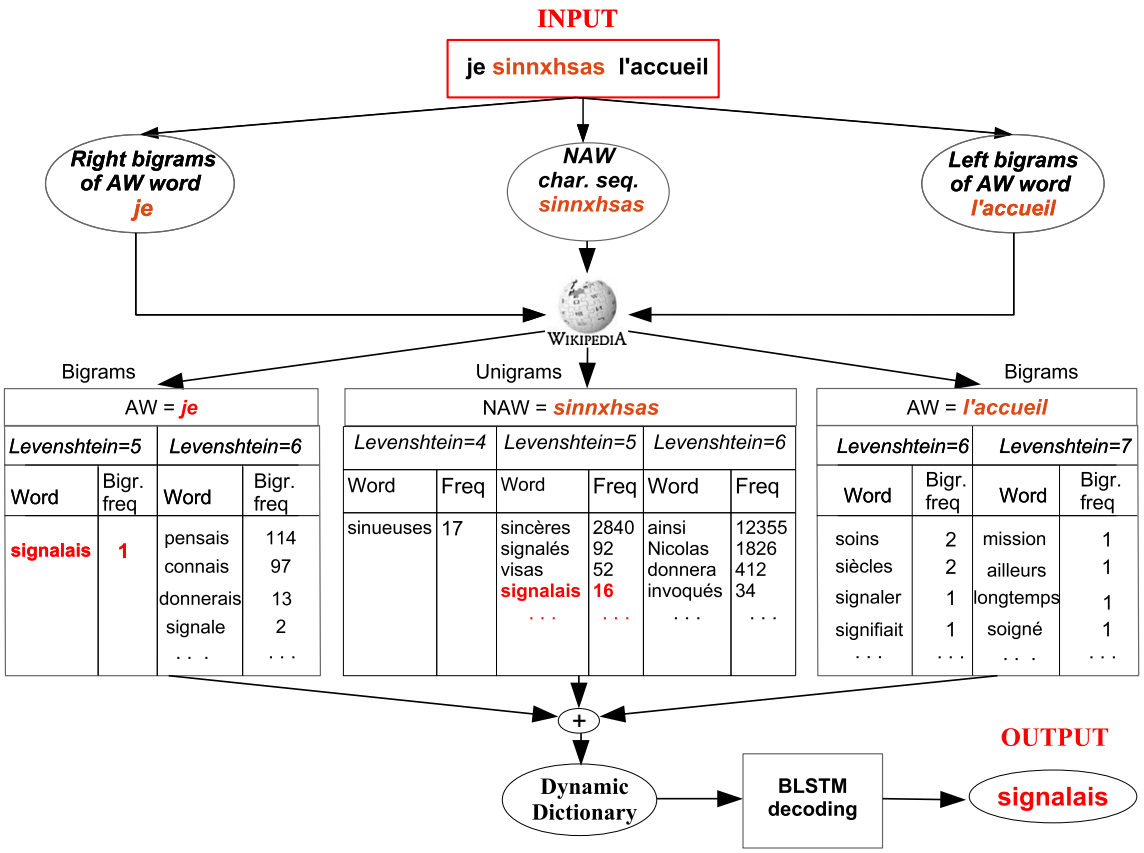


Fig. 8 Dynamic dictionary construction for an NAW. Dictionary words are collected from unigram and bigram Wikipedia resources

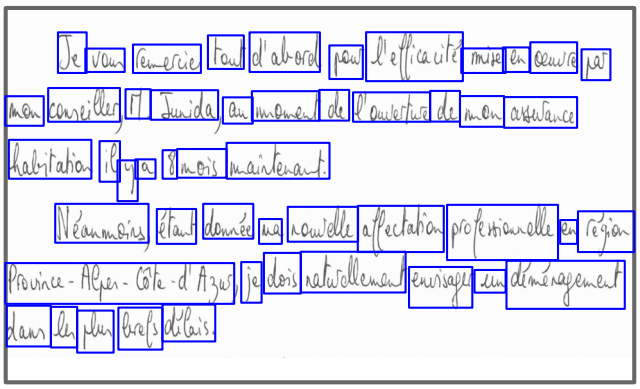
word sequence from bottom to top in order to collect the list of words corresponding to the left bigrams of NAWs whose neighbor on their right is an AW.

An example of the processing algorithm for the sequence AW–NAW–AW *"je signalais l'accueil"* is shown in Fig. 8. The character sequence c, output by the filler model for the NAW word, is *"sinnxhsas"*. Running the word sequence from top to bottom, the right bigrams for word *"je"* are retrieved. Running the sequence from bottom to top, the left bigrams for word *"l'accueil"* are identified. For instance for the AW word "je", the most frequent words that have this AW on the left are: *"ne"*, *"suis"*, *"me"*, *"vous"*, …, *"connaissais"*, …, *"signalais"*. For the AW word *"l'accueil"*, the most frequent words that have this AW on the right are: *"de"*, *"et"*, …, *"permettant"*,…, *"signifiait"*. We select only the most similar words with the NAW character sequence. In this case, the number of most similar words to the NAW character sequence, obtained from neighboring AWs is not enough. Therefore, the lists are expanded with the most similar unigrams. From these collected words, a dynamic dictionary is created and a second decoding is run with this adapted dictionary. Even though the ground truth word does not have a high document frequency in Wikipedia and its distance with the character sequence c is high, it can still be recovered. The NAW is replaced with the result of this decoding, in this case "signalais". For further iterations, this word will be considered as an AW. The algorithm is run until all NAWs are replaced by AWs.

An example of the use of the dynamic dictionary is given in Fig. 9. From the input word sequence, the result of the AW/NAW classification is provided: AWs are in black, NAWs in red. NAWs are still represented by the character sequence provided by the filler model. The final output is obtained by replacing the NAWs after decoding with dynamic dictionary. For this word sequence, the algorithm was iterated once, since all NAWs are surrounded by AWs.

7 Experiments

To assess the effectiveness of dynamic dictionary creation presented in Sect. 6, experiments are carried out with the RIMES [39] database. The metric used throughout all experiments is accuracy computed as the number of correctly recognized words divided by the testing set size. In the experiments described below, the results are case-insensitive (i.e., a = A), but accent errors are counted (i.e., a ≠ à).



(a) INPUT

(b) AW/NAW

Je vous remercie tout d'autand
pour efficacité mise en seule
par mon conseiller p Sunid
au moment de l'ouverture de mon
assurance habitation il n a 9
mois maintenant
Méaenmuoir étant donné ma
nouvelle afectation
professionnelle en région
Moince-hgr-céte dAgne je dois
ratllement envisager
un déménagement dans les plus
brefs délais

(c) OUTPUT

Je vous remercie tout d'abord
pour l'efficacité mise en seule
par mon conseiller p Suède
au moment de l'ouverture de mon
assurance habitation il n a 9
mois maintenant
Néanmoins étant donné ma
nouvelle affectation
professionnelle en région
Province-Alpes-Côte je dois
naturellement envisager un
déménagement dans les plus
brefs délais

Fig. 9 **a** Input word sequence, **b** AW/NAW classification (NAW in *red*), **c** output word sequence

7.1 RIMES database

The RIMES database (Reconnaissance et Indexation de données Manuscrites et de fax-similÉS/Recognition and Indexing of handwritten documents and faxes) [45] gathers different types of manuscripts written in French: correspondence, forms and faxes. It was created by the French Ministry of Defense to assess automatic handwriting recognition systems. RIMES has been used in evaluation campaigns since 2007 [39,46]. RIMES was created by asking volunteers to write letters in their own words related to scenarios such as bank account information, letters of complaint, payment difficulties or reminder letters. It brings together more than 12,500 handwritten documents written by around 1300 volunteers. The letters are written on white paper, in black ink, without guide lines, and are sampled at 300 dpi in gray scale (see Fig. 1).

For system implementation, we use the RIMES word and text databases used for the ICDAR 2011 French word recognition campaign [39]:

- The RIMES Word Dataset includes a training set of 51,738 word images, a validation set of 7464 word images and a testing set of 7776 word images. This dataset is provided as isolated words and contextual approaches, such as ours, cannot be applied to its testing set.
- The RIMES Text Dataset includes 1500 training text-blocks (11,329 text-line images) and 100 testing text-blocks (778 text-line images).

The original RIMES Text Dataset is not segmented into words and this segmentation is necessary for our experiments, in order to use contextual information, as described in Sect. 6.3. Therefore, from the text-blocks of the testing dataset, we have created a new dataset which we call the WS-RIMES-text database (word-segmented RIMES text database), which contains 5586 word images. This new set of word images includes the segmentation of the text-blocks into words. An HMM system [26] has been used for this purpose in a semi-automatic way (forced alignment).

The system is trained with the RIMES word training set using a lexicon of around 5000 different words, and calibrated with the RIMES word validation dataset, comprising a lexicon of around 1600 different words. The word-segmented WS-RIMES-text dataset was used for testing purposes. Around 7 % of the testing set words are not present in the training set and are OOVs.

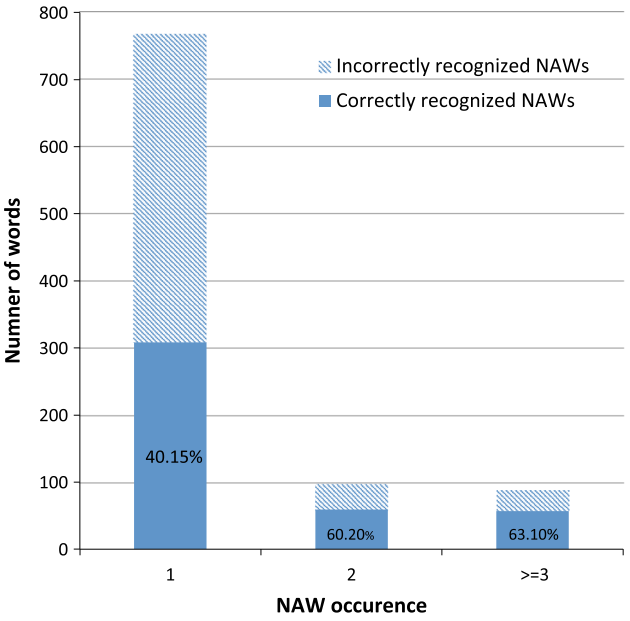


Fig. 10 NAW statistics—NAW occurrences and recognition accuracy in each case (NAW recognition accuracy and number of NAWs in the testing set database, as a function of word occurrence)

7.2 AW/NAW classification results

Figure 10 shows the histogram of NAW occurrences in the testing dataset. NAWs represent 17.1 % of the words in this set. Figure 10 also provides recognition accuracy in each bin, when NAWs are recognized using the dynamic dictionary-based approach.

A majority of the NAWs appear only once, and recognition accuracy reaches 40 % in these cases. Recognition rates are higher for more frequent NAWs that appear at least two or three times. This result is explained by the fact that the Wikipedia context of very rare NAWs is not robust enough.

7.3 Word recognition results

For each NAW word, a dynamic dictionary is created based on the domain-adapted Wikipedia bigrams and unigrams, with priority given to bigrams. In Table 1, we present results obtained with the *Static dictionary* and with our method *Dynamic dictionary*. In addition, we provide results that could be obtained if a perfect AW/NAW classification was available from an oracle, by considering all IVs as AWs and OOVs as NAWs. This is the *Dynamic dictionary, ideal AW/NAW* case. The ideal separation is used in order to highlight the maximum accuracy gain that could be expected using the proposed approach.

We provide the Wald confidence intervals computed as $\hat{p} \pm k * N^{-\frac{1}{2}} (\hat{p}(1-\hat{p}))^{\frac{1}{2}}$, where $\hat{p}$ is the proportion of well-recognized words, N the number of testing data, and k is the $100(1-\frac{\alpha}{2})$th percentile of the standard normal distribution, with a risk $\alpha = 5\,\%$. In Table 1, the improvement brought by the use of a Web resource exceeds 3% in absolute value (77.06 vs. 73.88 %) and is significant following the Wald test: The 77.06 % recognition rate is outside and over the *Static dictionary* approach. The difference between real and ideal AW/NAW separation shows that further progress is possible if this classification is improved.

Table 1 Recognition accuracies and dictionary coverage for static and dynamic dictionaries, with confidence intervals and the risk $\alpha = 5\,\%$. The size of the static dictionary is 4942, and the size of the dynamic dictionary is $k = 500$ words

Method	Coverage (%)	Accuracy (%)
Static dictionary	92.3	73.88 (72.73, 75.03)
Dynamic dictionary	93.5	77.06 (75.96, 78.16)
Dynamic dict., ideal AW/NAW	94.9	80.09 (79.04, 81.14)

A second set of experiments has been conducted to show the impact of dynamic dictionary creation on accuracy and dictionary coverage when different percentages of OOVs are considered. Starting with a 7.1 % rate of OOVs in the testing dataset, the least frequent words from the testing set are eliminated from the static dictionary, each time by 10 %, until OOVs represent approximately 55 % of the testing set. The sizes of the new static dictionaries are 4506, 4204, 4065, 3999 and 3966 words, corresponding to 15, 25, 35, 45 and 55 % of OOV words, respectively. Results are shown in Fig. 11 for all percentages of OOVs, when decoding with the corresponding static dictionary or using dynamic dictionaries. As expected, the overall recognition accuracy decreases for both cases as the OOV rate increases but it decreases less when using dynamic dictionaries. For instance, when approximately 55 % OOVs are present in the testing set, the recognition accuracy is equal to 60.66 % and only 43.62 % with the static dictionary. The improvement is thus greater than 17 % in absolute value. This improvement is even greater when the ideal AW/NAW classification is used: 22.5 %.

Figure 12 plots dictionary coverage as a function of OOV proportion in the testing set. When only 7 % of OOVs are present in the testing set, the coverages for the static dictionary and the dynamic dictionary are very similar due to AW/NAW misclassifications. When dynamic dictionaries are used, the coverage is always greater than when static dictionaries are used, because missing words are retrieved from the external resource. When the number of OOVs increases, both dictionary coverages decrease, but the coverage of the dynamic dictionary decreases less than that of the static dictionary. In the case the OOV rate is equal to 55 %, the coverage improvement brought by the dynamic dictionary is greater than 20 % .

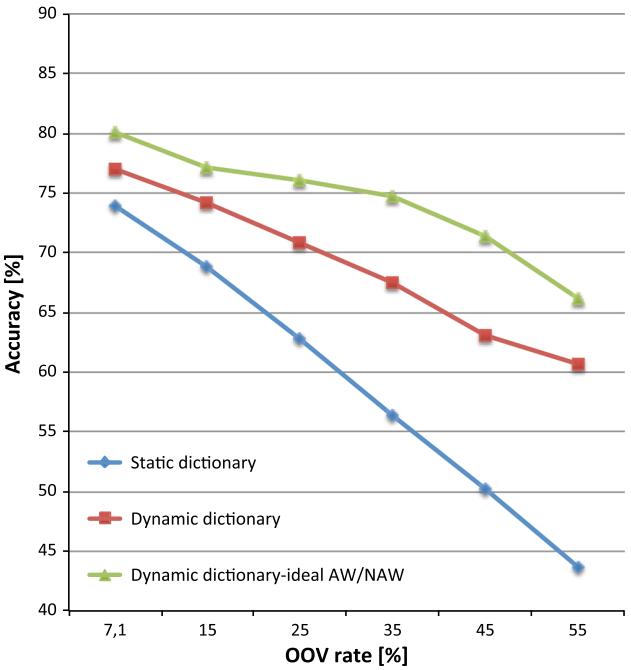


Fig. 11 Word recognition accuracy as a function of percentage of OOVs. The initial size of static dictionary is 4942 words

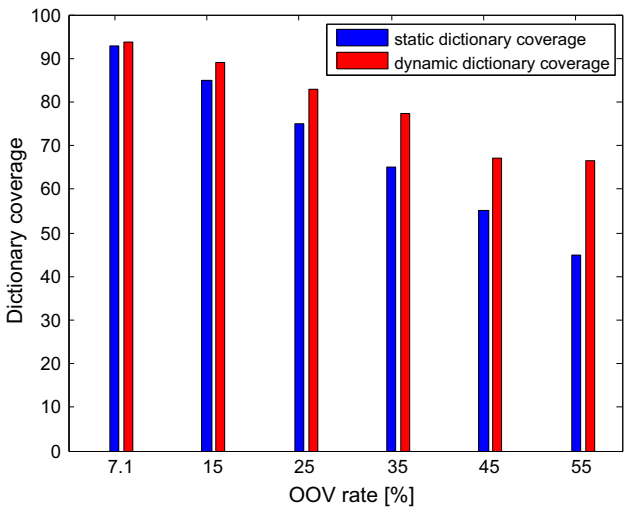


Fig. 12 Coverage of static and dynamic dictionaries for different percentages of OOV words

Figure 13 provides examples of recognition errors. The first example combines misrecognition of an AW word in the neighborhood of an NAW, with errors within the Wikipedia corpus. The AW word "vie" has been misrecognized as "ne". Unexpectedly, the right bigrams for word "ne" include word "can" even if "can" is not a French word and the transition "ne → can" does not exist in the language. However, "can" belongs to the French Wikipedia corpus. This can be explained by the fact that in the case of Wikipedia, different persons can edit articles and, as such, the online encyclopedia

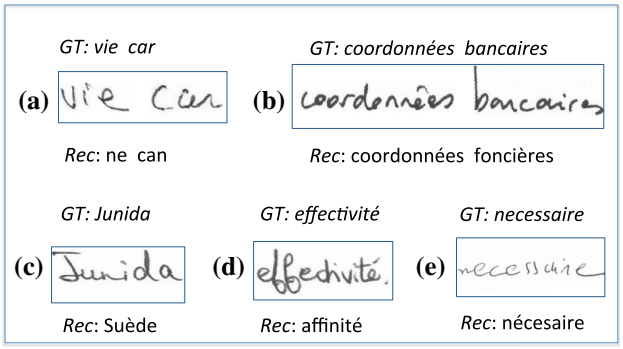


Fig. 13 Recognition error examples. The input image is provided, along with the ground truth (GT) and the recognized word (Rec)

is a representative example of crowd sourcing. The articles may contain though words that come from other languages or misspelled words (misprints, typos or syntax errors). Thus, string "can" has been introduced in the dynamic dictionary for the input image "car" shown in Fig. 13a and a confusion has occurred. The second error example is due to NAW sequences. When two NAWs follow each other, each NAW is recognized in isolation taking into consideration only one context, left other right. The linguistic relevance of the succession of the recognized words is currently not checked. In Fig. 13b, the NAW word sequence "coordonnées bancaires" (i.e., bank account details) has been recognized as "coordonnées foncières" since each NAW has been recognized separately. However, this word sequence does not exist in French ("foncières" meaning "property").

Recognition errors may occur when an input word is an OOV that is not included in the Wikipedia corpus. This happens for family names or words not commonly used. Even when these words are classified as NAWs, the dynamic dictionary cannot include the correct word. Thus the recognized words are similar words from Wikipedia. In Fig. 13c, "Junida" is a family name which has been recognized as "Suède" (Sweden in French), while word "effectivité" in Fig. 13d is an uncommon word in French which has been recognized as "affinité" (affinity in French).

The RIMES corpus itself can be a source of error. Since RIMES documents are written by volunteers using their own words, they contain misspelling errors. Typical examples are "addresser" (instead of adresser) or "courier" (instead of courrier). Around 3 % of the words from the static dictionary (training lexicon) are misspelled. Thus, if the spelling error is present in the training corpus, it will be introduced in the static dictionary. Therefore, during the decoding step, the erroneously spelled word may be chosen instead of the correct word such as the word "nécesaire" in Fig. 13e.

The relative improvement brought by dynamic dictionaries over static dictionaries is higher when OOV proportion increases, or equivalently, when static dictionary coverage decreases. Thus, the experiment presented in the following

Table 2 Recognition accuracies for a system without any initial static dictionary (only filler model-based recognition)—*BLSTM filler*. Dynamic dictionaries use Web resources at N-gram levels—*BLSTM filler + dyn. dict* $N = 1$ only (unigrams) or *BLSTM filler + dyn. dict* $N = 1, 2$ (unigrams and bigrams). Confidence intervals are provided for risk $\alpha = 5\,\%$

Method	Accuracy (%)
BLSTM filler	44.75 (43.45, 46.05)
BLSTM filler + dyn. dict (N-gram, $N = 1$).	67.66 (66.43, 68.89)
BLSTM filler + dyn dict. (N-gram, $N = 1, 2$)	69.08 (67.87, 70.29)

section consists of using no static dictionary (100 % OOVs, 0 % coverage) and Web-based dynamic dictionaries only.

7.4 Recognition without any initial static dictionary

These experiments are conducted in order to assess recognition accuracy starting with no static dictionary.

In the first experiment, the *BLSTM filler* is used in isolation, and the accuracy is 44.75 % (see Table 2). In the second experiment, all words are initially included in the NAW class. Thus, dynamic dictionaries are built from Web resources at unigram level only. The recognition accuracy *(BLSTM filler + dyn. dict, N-gram,* $N = 1$) at this step is 67.66 %. This result already improves the BLSTM filler by more than 23 % in absolute value. After this first iteration, some of the words are classified as AWs and bigrams can be included in dynamic dictionaries. The recognition process continues until no NAW is left. Recognition accuracy is further improved, brought by the use of bigrams *(BLSTM filler + dyn. dict, N-gram,* $N = 1, 2$).

If we now compare this result with the one using a static dictionary (Table 1, *Static dictionary*), the performance drop is less than 5 %.

This experiment is interesting on the one hand, because it highlights the discriminative power of RNNs and, on the other hand, because it shows that interesting results are obtained without using a static dictionary. Results from previous works on open vocabularies [1,47] show a larger gap between approaches which exploit only filler models and approaches which build a static dictionary. Here, this gap is much smaller and the combination of filler models and of dynamic dictionaries can be considered as a first step toward building high-performance open-vocabulary systems.

7.5 Recognition with large static dictionaries

Another set of experiments was performed in order to show the impact of the static dictionary size on the recognition rate. Instead of enlarging the dictionary dynamically, unigrams from the domain-adapted dictionary are gradually included into the static dictionary, according to their frequency.

Table 3 Recognition with large size static dictionaries created by including Wikipedia unigrams

Static dictionary size (k)	Coverage (%)	Accuracy (%)
5	92.3	73.88 (72.73, 75.03)
10	93.8	71.41 (70.24, 72.58)
20	95.9	71.10 (69.92, 72.28)
30	96.3	69.85 (68.66, 71.04)
40	96.7	69.28 (68.08, 70.48)
50	96.9	68.99 (67.79, 70.19)
75	97.3	60.84 (59.57, 62.11)

When the entire Wikipedia dictionary is used for decoding, together with the training dictionary (around 75,000 words), the system's performance drops to 60.84 %. This means around 9 % less than a system that uses no static dictionary at the beginning and dynamic dictionaries in a second phase (*BLSTM filler + dyn. dict, N-grams,* $N = 1, 2$) (see Sect. 7.4). This shows that as the dictionary is growing, confusions become prevalent and the overall quality of results decreases. Accuracies and dictionary coverages are shown in Table 3.

7.6 Discussion

Other results are available for the RIMES word testing database, including those of the ICDAR 2011 [39] French Handwriting Recognition Evaluation Campaign. However, these results are not easily comparable to ours. First, our testing set, WS-Rimes-text, is obtained by the segmentation of the RIMES text-lines into words, while the testing sets used at ICDAR 2011 are either line-based or word-based but without accounting for the lexical order of words. Second, the dictionary used for the word ICDAR 2011 competition includes the words of the training and testing sets, and therefore it does not contain any OOV word, a situation which is not often encountered in real-life applications. Third, the best results at ICDAR 2011 are obtained with relatively complex classifier combinations. Such combinations would probably have a positive effect on the performance of our proposed approach, but are out of focus here.

8 Conclusion and future work

The processing of OOV words is a central challenge in handwriting recognition. In our work, we propose an approach for robust handwriting recognition based on the combination of static and dynamic dictionaries. While recognizing a text, the proposed method builds on the fly dynamic dictionaries for words labeled as non-anchors, which are words for which we are not confident of the recognition results. There-

fore, and after a first recognition pass, recognized words are classified as anchors and non-anchors based on their recognition score and on the similarity between two recognition results represented as sequence of letters and obtained when the same recognizer is applied with and without a vocabulary. We propose to exploit Wikipedia, a large-scale Web lexical and linguistic resource to build the dynamic dictionaries associated with each non-anchor word. The proposed approach makes use of two criteria to select the words from Wikipedia to include in the dynamic dictionary. First, the words to be included in the dynamic dictionary need to be similar to the sequence of letters decoded for the non-anchor word when using a filler recognizer (recognition without vocabulary). Second, the words need to be relevant linguistically in the context of the non-anchor word. The proposed method performs better than a baseline system that uses a static dictionary. For the testing set of RIMES, we start from an initial static dictionary yielding 7% of OOV words. The accuracy obtained is 73.88 % when the static dictionary is used and reaches 77.06 % when our method is applied creating dynamic dictionaries for non-anchor words. The behavior of dynamic dictionary-based recognition in the presence of a larger number of OOVs is assessed by progressively decreasing the static vocabulary size. Naturally, a performance drop appears for both static and dynamic dictionaries, but results for dynamic dictionary-based recognition decrease more slowly due to the influence of the context and the exploitation of external resources, showing thereby higher robustness. Equally interesting, we show that dynamic dictionaries can be used to significantly reduce the performance gap between approaches that exploit static dictionaries and open-vocabulary systems.

Future work will be focused on the following points:

- The recognition of special cases of OOV words (codes, dates, telephone numbers, etc.), which are not recovered here, since Web resources may be less adapted to this task. Dedicated classifiers will be added to the system to further improve performance.
- Currently, only the immediate neighborhood of candidate words is exploited. A larger number of neighbors will be considered in order to make the context more robust.
- The methods introduced here are built on top of word-based segmentation of documents. Switching to text-line images would probably result in a higher recognition rate of anchor words, thus improving context reliability.
- Improving the AW/NAW classification method is important, given the potential increase in performance when the ideal separation is used. Since the two score distributions are superimposed, other separation criteria need to be investigated. In addition to the recognition scores, the language model could give cues for improving the detection of non-anchor words.
- Finally, the way we are building dynamic dictionaries can be viewed as an indexing problem. We will investigate building dynamic dictionaries by searching in an external database the documents corresponding to the anchor words. The words of those documents would form the dynamic dictionary.

Acknowledgments Authors wish to acknowledge ITESOFT/YOOZ which has supported this work by funding C. Oprean toward her PhD. We would also like to thank Pascal Vaillant of Paris 13 University for fruitful discussions on NLP approaches, as well as Alex Graves, Marcus Liwicki, Volkmar Frinken and Andreas Fischer for making available the RNN library.

References

1. Brakensiek, A., Willett, D., Rigoll, G.: Unlimited vocabulary script recognition using character N-grams. In: DAGM, pp. 436–443 (2000)
2. Bazzi, I., Schwartz, R.M., Makhoul, J.: An omnifont open-vocabulary OCR system for english and arabic. IEEE Trans. Pattern Anal. Mach. Intell. **21**(6), 495–504 (1999)
3. Hamdani, M., El-Desoky Mousa, A., Ney, H.: Open vocabulary arabic handwriting recognition using morphological decomposition. In: ICDAR, pp. 280–284 (2013)
4. Parada, C., Sethy, A., Dredze, M., Jelinek, F.: A spoken term detection framework for recovering out-of-vocabulary words using the web. In: INTERSPEECH (2010)
5. Oger, S., Popescu, V., Linarés, G.: Using the world wide web for learning new words in continuous speech recognition tasks: two case studies. In: SPECOM (2009)
6. Kaufmann, G., Bunke, H., Hadorn, M.: Lexicon reduction in an HMM-framework based on quantized feature vectors. In: ICDAR, pp. 1097–1101 (1997)
7. Guillevic, D., Nishiwaki, D., Yamada, K.: Word lexicon reduction by character spotting. In: IWFHR, pp. 373–382 (2000)
8. Powalka, R.K., Sherkat, N., Whitrow, R.J.: Word shape analysis for a hybrid recognition system. Pattern Recogn. **30**(3), 421–445 (1997)
9. Seni, G., Srihari, R.K., Nasrabadi, N.M.: Large vocabulary recognition of on-line handwritten cursive words. IEEE Trans. Pattern Anal. Mach. Intell. **18**(7), 757–762 (1996)
10. Leroy, A.: Lexicon reduction based on global features for on-line handwriting. In: IWFHR, pp. 431–440 (1994)
11. Vinciarelli, A.: Noisy text categorization. IEEE Trans. Pattern Anal. Mach. Intell. **27**(12), 1882–1895 (2005)
12. Milewski, R., Govindaraju, V., Bhardwaj, A.: Automatic recognition of handwritten medical forms for search engines. IJDAR **11**(4), 203–218 (2009)
13. Farooq, F., Chandalia, G., Govindaraju, V.: Lexicon reduction in handwriting recognition using topic categorization. In: DAS, pp. 369–375 (2008)
14. Farooq, F., Bhardwaj, A., Govindaraju, V.: Using topic models for ocr correction. IJDAR **12**(3), 153–164 (2009)
15. Whitelaw, C., Hutchinson, B., Chung, G., Ellis, G.: Using the web for language independent spellchecking and autocorrection. In: EMNLP, pp. 890–899 (2009)
16. Soricut, R., Brill, E.: Automatic question answering using the web: beyond the factoid. Inf. Retrieval **9**(2), 191–206 (2006)
17. Rigau, G., Magnini, B., Agirre, E., Vossen, P., Carroll, J.: Meaning: a roadmap to knowledge technologies. In: COLING-02 on A roadmap for computational linguistics, pp. 1–7 (2002)

18. Grefenstette, G.: The World Wide Web as a resource for example-based machine translation tasks. In: Translating and the Computer 21: Proceedings of the 21st International Conference on Translating and the Computer (1999)
19. Cao, Y.: Base noun phrase translation using web data and the EM algorithm. In: Proceedings of CoLing, pp. 127–133 (2002)
20. Oger, S., Popescu, V., Linarés, G.: Using the world wide web for learning new words in continuous speech recognition tasks: two case studies. In: SPECOM (2009)
21. Keller, F., Lapata, M.: Using the web to obtain frequencies for unseen bigrams. Comput. Linguistics **29**(3), 459–484 (2003)
22. Adler, M., Goldberg, Y., Gabay, D., Elhadad, M.: Unsupervised lexicon-based resolution of unknown words for full morphological analysis. In: ACL, pp. 728–736 (2008)
23. Umansky-Pesin, S., Reichart, R., Rappoport, A.: A multi-domain web-based algorithm for POS tagging of unknown words. Beijing, pp. 1274–1282 (2010)
24. Taghva, K., Agarwal, S.: Utilizing web data in identification and correction of OCR errors. In: Proceedings of DRR (2014)
25. Feild, J. L., Learned-Miller, E. G.: Improving open-vocabulary scene text recognition. In: ICDAR, pp. 604–608 (2013)
26. Oprean, C., Likforman-Sulem, L., Popescu, A., Mokbel, C.: Using the web to create dynamic dictionaries in handwritten out-of-vocabulary word recognition. In: ICDAR, pp. 989–993 (2013)
27. Vinciarelli, A., Luettin, J.: A new normalization technique for cursive handwritten words. Pattern Recogn. Lett. **22**(9), 1043–1050 (2001)
28. Bianne-Bernard, A.-L., Menasri, F., El-Hajj, R., Mokbel, C., Kermorvant, C., Likforman-Sulem, L.: Dynamic and contextual information in HMM modeling for handwritten word recognition. IEEE PAMI **99**(10), 2066–2080 (2011)
29. Oprean, C., Likforman-Sulem, L., Mokbel, C.: Handwritten word preprocessing for database adaptation. In: DRR XX, pp. 808–865 (2013)
30. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**, 2673–2681 (1997)
31. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: Kolen, J., Kremer, S. (eds.) Field Guide to Dynamical Recurrent Networks. IEEE Press, New York (2001)
32. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. PAMI **31**(5), 855–868 (2009)
33. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. Neural Netw. **1**(4), 339–356 (1988)
34. Williams, R. J., Zipser, D.: Backpropagation: theory, architecture and applications. In: Chauvin, Y., Rumelhart, D.E. (eds.) Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity, pp. 433–486. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1995)
35. Graves, A., Fernández, S., Gomez, F.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: ICML, pp. 369–376 (2006)
36. Frinken, V., Fischer, A., Manmatha, R., Bunke, H.: A novel word spotting method based on recurrent neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 211–224 (2012)
37. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
38. Singhal, A.: Modern information retrieval: a brief overview. IEEE Data Eng. Bull. **24**(4), 35–43 (2001)
39. Grosicki, E., El-Abed, H.: ICDAR 2011-French handwriting recognition competition. In: ICDAR, pp. 1459–1463 (2011)
40. Hayamizu, S., Itou, K., Tanaka, K.: Detection of unknown words in large vocabulary speech recognition. In: EUROSPEECH (1993)
41. White, C. M., Zweig, G., Burget, L., Schwarz, P., Hermansky, H.: Confidence estimation, OOV detection and language ID using phone-to-word transduction and phone-level alignments. In: ICASSP, pp. 4085–4088 (2008)
42. Burget, L., Schwarz, P., Matějka, P., Hannemann, M., Rastrow, A., White, C., Khudanpur, S., Heřmanský, H., Černocký, J.: Combination of strongly and weakly constrained recognizers for reliable detection of OOVs. In: ICASSP (2008)
43. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Phys. Doklady **10**, 707 (1966)
44. Damerau, F.: A technique for computer detection and correction of spelling errors. Commun. ACM **7**, 171–176 (1964)
45. Grosicki, E., Carré, M., Geoffrois, E., Augustin, E., Preteux, F.: La campagne d'évaluation RIMES pour la reconnaissance de courriers manuscrits. In: CIFED (2006)
46. Grosicki, E., Abed, H. E.: ICDAR 2009 handwriting recognition competition. In: ICDAR (2009)
47. Brakensiek, A., Rottland, J., Kosmala, A., Rigoll, G.: Off-line handwriting recognition using various hybrid modeling techniques and character n-grams. In: IWFHR, pp. 343–352 (2000)