4 Related Work

Our system is based on the idea of [2] to learn to align and transcribe for machine translation. It is achieved by coupling an encoder of the input signal and a decoder predicting language tokens with an attention mechanism, which selects from the encoded signal the relevant parts for the next prediction.

It bears many similarity with the attention-based models for speech recognition [8, 10]. Indeed, we want to predict text from a sensed version of natural language (audio in speech recognition, image of handwritten text here). As for speech recognition, we need to deal with long sequences. Our network also has LSTM recurrences, but we use MDLSTM units to handle images, instead of bi-directional LSTMs. This is a different way of handling images, compared with the attention-based systems for image captioning for example [9, 29]. Besides the MDLSTM attention, the main difference in our architecture is that we do not input the previous character to predict the next one, so it is also quite different from the RNN transducers [13].

Contrary to some attention models like DRAW [16] or spatial transformer networks [17], our model does not select and transform a part of the input by interpolation, but only weights the feature vectors and combine them with a sum. We do not explicitely predict the coordinates of the attention, as done in [1].

In similar models of attention, the weights are either computed from the content at each position individually (e.g. in [8, 29]), from the location of the previous attention (e.g. in [14, 15]) or from a combination of both (e.g. in [10, 15]). In our model, the content of the whole image is explicitely taken into account to predict the weight at every position, and the location is implicitely considered through the MDLSTM recurrences.

Finally, although attention models have been applied to the recognition of sequences of symbols (e.g. in [1, 26] for MNIST or SVHN digits, and [20, 25] for scene text OCR on cropped words), we believe that we present the first attempt to recognize multiple lines of cursive text without an explicit line segmentation.

5 Experiments

5.1 Experimental Setup

We carried out the experiments on the popular IAM database, described in details in [22], consisting of images of handwritten English text documents. They correspond to English texts exctracted from the LOB corpus. 657 writers produced between 1 and 59 handwritten documents. The training set comprises 747 documents (6,482 lines, 55,081 words), the validation set 116 documents (976 lines, 8,895 words) and the test set 336 documents (2,915 lines, 25,920 words). The texts in this database typically contain 450 characters in about nine lines. In 150 dpi images, the average character has a width of 20px.

The baseline corresponds to the architecture presented in Figure 1, with 4, 20 and 100 units in MDLSTM layers, 12 and 32 units in convolutional layers, and dropout after every MDLSTM as presented in [23]. The last linear layer has 80 outputs, and is followed by a collapse layer and a softmax normalization. In the attention-based model, the encoder has the same architecture as the baseline model, without the collapse and softmax. The attention network has 16 or 32 hidden LSTM units in each direction followed by a linear layer with one output. The state LSTM layer has 128 or 256 units, and the decoder is an MLP with 128 or 256 tanh neurons. The networks are trained with RMSProp [27] with a base learning rate of 0.001 and mini-batches of 8 examples. We measure the Character Error Rate (CER%), i.e. the edit distance normalized by the number of characters in the ground-truth.

5.2 The Usual Word and Line Recognition Tasks

We first trained the model to recognize words and lines. The inputs are images of several consecutive words from the IAM database. The encoder network has the standard architecture presented in Section 2, with dropout after each LSTM layer [23] and was pre-trained on IAM database with

4 Related Work

Our system is based on the idea of [2] to learn to align and transcribe for machine translation. It is achieved by coupling an encoder of the input signal and a decoder predicting language tokens with an attention mechanism, which selects from the encoded signal the relevant parts for the next prediction.

It bears many similarity with the attention-based models for speech recognition [8, 10]. Indeed, we want to predict text from a sensed version of natural language (audio in speech recognition, image of handwritten text here). As for speech recognition, we need to deal with long sequences. Our network also has LSTM recurrences, but we use MDLSTM units to handle images, instead of bi-directional LSTMs. This is a different way of handling images, compared with the attention-based systems for image captioning for example [9, 29]. Besides the MDLSTM attention, the main difference in our architecture is that we do not input the previous character to predict the next one, so it is also quite different from the RNN transducers [13].

Contrary to some attention models like DRAW [16] or spatial transformer networks [17], our model does not select and transform a part of the input by interpolation, but only weights the feature vectors and combine them with a sum. We do not explicitely predict the coordinates of the attention, as done in [1].

In similar models of attention, the weights are either computed from the content at each position individually (e.g. in [8, 29]), from the location of the previous attention (e.g. in [14, 15]) or from a combination of both (e.g. in [10, 15]). In our model, the content of the whole image is explicitely taken into account to predict the weight at every position, and the location is implicitely considered through the MDLSTM recurrences.

Finally, although attention models have been applied to the recognition of sequences of symbols (e.g. in [1, 26] for MNIST or SVHN digits, and [20, 25] for scene text OCR on cropped words), we believe that we present the first attempt to recognize multiple lines of cursive text without an explicit line segmentation.

5 Experiments

5.1 Experimental Setup

We carried out the experiments on the popular IAM database, described in details in [22], consisting of images of handwritten English text documents. They correspond to English texts exctracted from the LOB corpus. 657 writers produced between 1 and 59 handwritten documents. The training set comprises 747 documents (6,482 lines, 55,081 words), the validation set 116 documents (976 lines, 8,895 words) and the test set 336 documents (2,915 lines, 25,920 words). The texts in this database typically contain 450 characters in about nine lines. In 150 dpi images, the average character has a width of 20px.

The baseline corresponds to the architecture presented in Figure 1, with 4, 20 and 100 units in MDLSTM layers, 12 and 32 units in convolutional layers, and dropout after every MDLSTM as presented in [23]. The last linear layer has 80 outputs, and is followed by a collapse layer and a softmax normalization. In the attention-based model, the encoder has the same architecture as the baseline model, without the collapse and softmax. The attention network has 16 or 32 hidden LSTM units in each direction followed by a linear layer with one output. The state LSTM layer has 128 or 256 units, and the decoder is an MLP with 128 or 256 tanh neurons. The networks are trained with RMSProp [27] with a base learning rate of 0.001 and mini-batches of 8 examples. We measure the Character Error Rate (CER%), i.e. the edit distance normalized by the number of characters in the ground-truth.

5.2 The Usual Word and Line Recognition Tasks

We first trained the model to recognize words and lines. The inputs are images of several consecutive words from the IAM database. The encoder network has the standard architecture presented in Section 2, with dropout after each LSTM layer [23] and was pre-trained on IAM database with

4 Related Work

Our system is based on the idea of [2] to learn to align and transcribe for machine translation. It is achieved by coupling an encoder of the input signal and a decoder predicting language tokens with an attention mechanism, which selects from the encoded signal the relevant parts for the next prediction.

It bears many similarity with the attention-based models for speech recognition [8, 10]. Indeed, we want to predict text from a sensed version of natural language (audio in speech recognition, image of handwritten text here). As for speech recognition, we need to deal with long sequences. Our network also has LSTM recurrences, but we use MDLSTM units to handle images, instead of bi-directional LSTMs. This is a different way of handling images, compared with the attention-based systems for image captioning for example [9, 29]. Besides the MDLSTM attention, the main difference in our architecture is that we do not input the previous character to predict the next one, so it is also quite different from the RNN transducers [13].

Contrary to some attention models like DRAW [16] or spatial transformer networks [17], our model does not select and transform a part of the input by interpolation, but only weights the feature vectors and combine them with a sum. We do not explicitely predict the coordinates of the attention, as done in [1].

In similar models of attention, the weights are either computed from the content at each position individually (e.g. in [8, 29]), from the location of the previous attention (e.g. in [14, 15]) or from a combination of both (e.g. in [10, 15]). In our model, the content of the whole image is explicitely taken into account to predict the weight at every position, and the location is implicitely considered through the MDLSTM recurrences.

Finally, although attention models have been applied to the recognition of sequences of symbols (e.g. in [1, 26] for MNIST or SVHN digits, and [20, 25] for scene text OCR on cropped words), we believe that we present the first attempt to recognize multiple lines of cursive text without an explicit line segmentation.

5 Experiments

5.1 Experimental Setup

We carried out the experiments on the popular IAM database, described in details in [22], consisting of images of handwritten English text documents. They correspond to English texts exctracted from the LOB corpus. 657 writers produced between 1 and 59 handwritten documents. The training set comprises 747 documents (6,482 lines, 55,081 words), the validation set 116 documents (976 lines, 8,895 words) and the test set 336 documents (2,915 lines, 25,920 words). The texts in this database typically contain 450 characters in about nine lines. In 150 dpi images, the average character has a width of 20px.

The baseline corresponds to the architecture presented in Figure 1, with 4, 20 and 100 units in MDLSTM layers, 12 and 32 units in convolutional layers, and dropout after every MDLSTM as presented in [23]. The last linear layer has 80 outputs, and is followed by a collapse layer and a softmax normalization. In the attention-based model, the encoder has the same architecture as the baseline model, without the collapse and softmax. The attention network has 16 or 32 hidden LSTM units in each direction followed by a linear layer with one output. The state LSTM layer has 128 or 256 units, and the decoder is an MLP with 128 or 256 tanh neurons. The networks are trained with RMSProp [27] with a base learning rate of 0.001 and mini-batches of 8 examples. We measure the Character Error Rate (CER%), i.e. the edit distance normalized by the number of characters in the ground-truth.

5.2 The Usual Word and Line Recognition Tasks

We first trained the model to recognize words and lines. The inputs are images of several consecutive words from the IAM database. The encoder network has the standard architecture presented in Section 2, with dropout after each LSTM layer [23] and was pre-trained on IAM database with

4 Related Work

Our system is based on the idea of [2] to learn to align and transcribe for machine translation. It is achieved by coupling an encoder of the input signal and a decoder predicting language tokens with an attention mechanism, which selects from the encoded signal the relevant parts for the next prediction.

It bears many similarity with the attention-based models for speech recognition [8, 10]. Indeed, we want to predict text from a sensed version of natural language (audio in speech recognition, image of handwritten text here). As for speech recognition, we need to deal with long sequences. Our network also has LSTM recurrences, but we use MDLSTM units to handle images, instead of bi-directional LSTMs. This is a different way of handling images, compared with the attention-based systems for image captioning for example [9, 29]. Besides the MDLSTM attention, the main difference in our architecture is that we do not input the previous character to predict the next one, so it is also quite different from the RNN transducers [13].

Contrary to some attention models like DRAW [16] or spatial transformer networks [17], our model does not select and transform a part of the input by interpolation, but only weights the feature vectors and combine them with a sum. We do not explicitely predict the coordinates of the attention, as done in [1].

In similar models of attention, the weights are either computed from the content at each position individually (e.g. in [8, 29]), from the location of the previous attention (e.g. in [14, 15]) or from a combination of both (e.g. in [10, 15]). In our model, the content of the whole image is explicitely taken into account to predict the weight at every position, and the location is implicitely considered through the MDLSTM recurrences.

Finally, although attention models have been applied to the recognition of sequences of symbols (e.g. in [1, 26] for MNIST or SVHN digits, and [20, 25] for scene text OCR on cropped words), we believe that we present the first attempt to recognize multiple lines of cursive text without an explicit line segmentation.

5 Experiments

5.1 Experimental Setup

We carried out the experiments on the popular IAM database, described in details in [22], consisting of images of handwritten English text documents. They correspond to English texts exctracted from the LOB corpus. 657 writers produced between 1 and 59 handwritten documents. The training set comprises 747 documents (6,482 lines, 55,081 words), the validation set 116 documents (976 lines, 8,895 words) and the test set 336 documents (2,915 lines, 25,920 words). The texts in this database typically contain 450 characters in about nine lines. In 150 dpi images, the average character has a width of 20px.

The baseline corresponds to the architecture presented in Figure 1, with 4, 20 and 100 units in MDLSTM layers, 12 and 32 units in convolutional layers, and dropout after every MDLSTM as presented in [23]. The last linear layer has 80 outputs, and is followed by a collapse layer and a softmax normalization. In the attention-based model, the encoder has the same architecture as the baseline model, without the collapse and softmax. The attention network has 16 or 32 hidden LSTM units in each direction followed by a linear layer with one output. The state LSTM layer has 128 or 256 units, and the decoder is an MLP with 128 or 256 tanh neurons. The networks are trained with RMSProp [27] with a base learning rate of 0.001 and mini-batches of 8 examples. We measure the Character Error Rate (CER%), i.e. the edit distance normalized by the number of characters in the ground-truth.

5.2 The Usual Word and Line Recognition Tasks

We first trained the model to recognize words and lines. The inputs are images of several consecutive words from the IAM database. The encoder network has the standard architecture presented in Section 2, with dropout after each LSTM layer [23] and was pre-trained on IAM database with

4 Related Work

Our system is based on the idea of [2] to learn to align and transcribe for machine translation. It is achieved by coupling an encoder of the input signal and a decoder predicting language tokens with an attention mechanism, which selects from the encoded signal the relevant parts for the next prediction.

It bears many similarity with the attention-based models for speech recognition [8, 10]. Indeed, we want to predict text from a sensed version of natural language (audio in speech recognition, image of handwritten text here). As for speech recognition, we need to deal with long sequences. Our network also has LSTM recurrences, but we use MDLSTM units to handle images, instead of bi-directional LSTMs. This is a different way of handling images, compared with the attention-based systems for image captioning for example [9, 29]. Besides the MDLSTM attention, the main difference in our architecture is that we do not input the previous character to predict the next one, so it is also quite different from the RNN transducers [13].

Contrary to some attention models like DRAW [16] or spatial transformer networks [17], our model does not select and transform a part of the input by interpolation, but only weights the feature vectors and combine them with a sum. We do not explicitely predict the coordinates of the attention, as done in [1].

In similar models of attention, the weights are either computed from the content at each position individually (e.g. in [8, 29]), from the location of the previous attention (e.g. in [14, 15]) or from a combination of both (e.g. in [10, 15]). In our model, the content of the whole image is explicitely taken into account to predict the weight at every position, and the location is implicitely considered through the MDLSTM recurrences.

Finally, although attention models have been applied to the recognition of sequences of symbols (e.g. in [1, 26] for MNIST or SVHN digits, and [20, 25] for scene text OCR on cropped words), we believe that we present the first attempt to recognize multiple lines of cursive text without an explicit line segmentation.

5 Experiments

5.1 Experimental Setup

We carried out the experiments on the popular IAM database, described in details in [22], consisting of images of handwritten English text documents. They correspond to English texts exctracted from the LOB corpus. 657 writers produced between 1 and 59 handwritten documents. The training set comprises 747 documents (6,482 lines, 55,081 words), the validation set 116 documents (976 lines, 8,895 words) and the test set 336 documents (2,915 lines, 25,920 words). The texts in this database typically contain 450 characters in about nine lines. In 150 dpi images, the average character has a width of 20px.

The baseline corresponds to the architecture presented in Figure 1, with 4, 20 and 100 units in MDLSTM layers, 12 and 32 units in convolutional layers, and dropout after every MDLSTM as presented in [23]. The last linear layer has 80 outputs, and is followed by a collapse layer and a softmax normalization. In the attention-based model, the encoder has the same architecture as the baseline model, without the collapse and softmax. The attention network has 16 or 32 hidden LSTM units in each direction followed by a linear layer with one output. The state LSTM layer has 128 or 256 units, and the decoder is an MLP with 128 or 256 tanh neurons. The networks are trained with RMSProp [27] with a base learning rate of 0.001 and mini-batches of 8 examples. We measure the Character Error Rate (CER%), i.e. the edit distance normalized by the number of characters in the ground-truth.

5.2 The Usual Word and Line Recognition Tasks

We first trained the model to recognize words and lines. The inputs are images of several consecutive words from the IAM database. The encoder network has the standard architecture presented in Section 2, with dropout after each LSTM layer [23] and was pre-trained on IAM database with