A Hybrid BLSTM-HMM for spotting Regular Expressions

Gautier Bideault[1], Luc Mioulet[1], Clément Chatelain [2] and Thierry Paquet [1]
[1]*Laboratoire LITIS - EA 4108, Universite de Rouen, FRANCE 76800*
[2]*Laboratoire LITIS - EA 4108, INSA Rouen, FRANCE 76800*

Keywords: Handwriting Recognition, Regular Expression Spotting, REGEX, HMM, BLSTM, Word Spotting

Abstract: This article concerns the spotting of regular expressions (REGEX) in handwritten documents using a hybrid model. Spotting REGEX in a document image allow to consider further extraction tasks such as document categorization or named entities extraction. Our model combines state of the art BLSTM recurrent neural network for character recognition and segmentation with a HMM model able to spot the desired sequences. Our experiments on a public handwritten database show interesting results.

1 INTRODUCTION

Detecting Regular expression (REGEX) in handwritten documents can be useful for finding sub-string which are relevant for a further higher level information extraction task. It consists in detecting patterns sequence of characters that obey certain rules described using meta models such as lower cases (#[a-z]#), upper cases (#[A-Z]#) or Digits (#[0-9]#). For example, a system of that kind could spot entities. For example, spotting date (#[0-9]{2}/[0-9]{2}/[0-9]{4}#), first name (#[A-Z][a-z]*#), ZIP code and city name of a french postal address (#[0-9]{5} [A-Z]*#). The extraction of these information allow to consider high level processing stages such as document categorisation, customer identification, Named Entity detection, etc.

The spotting of regular expression is a common task on electronical documents, using Natural Language Processing methods (Hosoya and Pierce, 2001; Dengel and Klein, 2002). In this case, the REGEX spotting is rather straightforward as it consists in applying exact string matching methods on the ASCII text. When dealing with document images, a recognition step is needed in order to produce the ASCII transcription before processing the input data. The trouble is that this recognition step is subject to errors and uncertainty, making the string matching problematic. Some attempts have been made on printed documents (Spitz, 1995; .Spitz, 1997). In these works, an OCR is applied on the whole document before applying the regular expression spotting step based on a set of rules that performs an exact matching. In spite of OCR errors, the system provides acceptable performance (Average Precision of 82% and 72% of Recall).

However, only a few works concerning regular expression spotting in handwritten documents. The reason is that exact matching methods can not overcome the frequent recognition errors due to the intrinsic difficulties of recognizing handwriting. Therefore, in order to cope with these errors, inexact matching method should be carried out. This can be performed using statistical sequence models such as HMM. Some works have been published within this framework, proposing pattern spotting such as dates (Morita et al., 2003) and numerical fields(Chatelain et al., 2006; Chatelain et al., 2008) that involve meta models of characters, namely digits. However, these HMM based approach are limited to very specific fields.

A more generic approach for REGEX spotting in handwritten documents has been addressed using pure HMM approach(Kessentini et al., 2013) , but led to moderate results (see section 5 for detailed results and comparison). This article presents a REGEX spotting system for handwritten documents. It is based on a combination of HMM statistical sequence model with the state-of-the-art BLSTM neural network. Our alternative hybrid BLSTM/HMM model enables us to benefit from both strong local discrimination, and the generative sequence ability of the HMM.

This paper is organized as follows: first a review of word and REGEX spotting is given in section 2, then we present our REGEX spotting system based on a hybrid BLSTM/HMM in section 3. Section 4 is devoted to the experimental setup and results on both word spotting and regular expression spotting tasks carried out on the RIMES database(Grosicki and El Abed, 2009).

2 RELATED WORK

As a REGEX can match sequences with variable length and characters, a REGEX spotting task can be assimilated to a word spotting task where the word belongs to a lexicon which contains all the character string variations admissible by the REGEX. The less constrained the REGEX, the larger the size of the lexicon. Relaxing those constraints makes the REGEX spotting task more complex especially when considering handwritten document images. As regular expression spotting shares many aspects in common with word spotting we now briefly introduce the related works concerning word spotting approaches.

Word spotting in document images has received a lot of attention these last years. Systems proposed in the literature are divided into two main categories : Image based and recognition based systems. The first one, also known as *query-by-example*, operates through the image representation of the keywords(Rath and Manmatha, 2003; Cao and Govindaraju, 2007; Adamek et al., 2007; Rusinol et al., 2011; Rodríguez-Serrano et al., 2009). Such systems are therefore limited to deal with omni-writer handwriting and require to get an image of the query. The second kind of approaches, also known as *query-by-string* methods, deals with the ASCII representation of the keywords(Rodríguez-Serrano and Perronnin, 2009; Frinken et al., 2012; Thomas et al., 2010; Fischer et al., 2012; Wshah et al., 2012).Moving from the image representation to the ascii representation of the query is performed through a recognition stage. These systems are suitable for omni-writer handwriting and can be used with any string query of any size. In this context, many works have focused on several variants of Hidden Markov Models (HMMs) to process this intrinsically sequential problem (Wshah et al., 2012).

State-of-the-art recognition-based approaches are based on a line of text models (Thomas et al., 2010; Fischer et al., 2012; Wshah et al., 2012). The line model generally contains a model of the target word, combined with filler models that describe the out-of-vocabulary words. For example in (Thomas et al., 2010), the authors present an alpha-numerical information extraction system on handwritten unconstrained documents. It relies on a global line modeling allowing a dual representation of the relevant and the irrelevant information. The acceptation or rejection of the matched keyword is controlled by the variation of a hyper-parameter in the HMM line model. A similar approach is presented in (Fischer et al., 2012). The line model is made of a left and right filler models surrounding the word model. The acceptation or rejection of the matched keyword is controlled by a text line score based on the likelihood ratio of the word text line and the filler text line model. However we know that HMM rely on strong observation independence assumptions and they perform poorly on high dimensional observations. Moreover, they have low discrimination capabilities between character classes due to their inherently generative modelization framework.

Recently a new approach based on recurrent neural networks has overcome these shortcomings. Bilateral Long Short Term Memory (BLSTM) architecture has demonstrated impressive capabilities for omni-writer handwriting recognition (Graves et al., 2008). Some primary applications of BLSTM to word spotting have also demonstrated promising results (Frinken et al., 2010; Wöllmer et al., 2009). In this system, the BLSTM is combined with the CTC layer which provides character class posterior probabilities. Then a token passing algorithm allows efficient decoding of the spotting line model. Very interesting results have been reported on the IAM Database (Frinken et al., 2010)[1].

In this paper, we combine the BLSTM-CTC architecture with a HMM based spotting line model. This two stage architecture is first evaluated for hadnwritten word spotting on the RIMES database. Then we explore some extensions of the system to Regular Expression spotting (REGEX Spotting). This model is described in the following section.

3 BLSTM-CTC/HMM SYSTEM

In this section, we describe our hybrid model for word and REGEX spotting. We first describe the BLSTM-HMM architecture that has been retained, then we present our word spotting model, based on standard state-of-the-art word spotting framework. And finally, we propose the adaptation of this model for REGEX spotting.

3.1 Character Recognition and Segmentation

The BLSTM-CTC is a complex Recurrent Neural Network able to manage long term dependencies thanks to its internal buffer structure. Each neuron is specialized to stop a specific character in the input signal. The recurrent architecture allows each neuron to take account of the previous activated neurons (character), possibly at multiple time step earlier in the input signal (thus modeling long term dependencies). This typical architecture allows to take account

[1] Average Precision of 88.15% and R-precision of 84.34%

of character bigrams in addition to the input signal to compute the activation of each neuron. The BLSTM is composed of two recurrent neural networks with Long Short Term Memory neural units. The first one processes the data from left to right whereas the second one proceeds in the reverse order. For each time step, decision is taken combining the two networks output, taking advantage of both left and right context. Such context is essential to have a certain knowledge of the surounding characters, because in most cases sequences of letters are constrained by the properties of the lexicon. The outputs of these two networks are then combined through a Softmax decision layer that provides character posterior probabilities in addition to a non decision class. This decision stage is called the Connectionist Temporal Classification (CTC)(Graves et al., 2006) that enables the labelling of unsegmented data.

These networks integrate special neural network units : Long Short Term Memory (Graves et al., 2006) (LSTM). LSTM neurons is composed of a memory cel, an input and three control gates. Each gates control the memory of the cell, i.e how a given input will affect the memory (input gates), if a new input should reset the memory cell (forget gate) and if the memory of the network should be presented to the following neuron (output gate). This system of control gates allows a very accurate control of the memory cell during the training step. A LSTM layer is fully recurrent, that is to say, the input and the three gates receive at each instant t the input signal at time t and the previous outputs (at time $(t-1)$).

This architecture has shown very impressive results on challenging data-sets dedicated to word recognition (Graves et al., 2009; Grosicki and El Abed, 2009) due to its efficient classification and segmentation ability. For these reasons, its efficiency to cope with the low level character identification appears also very promising for handwritten words and REGEX spotting, since such scenario is less constrained by lexicon properties.

The proposed BLSTM/HMM architecture has been chosen in order to take advantage of both generative and discriminative frameworks. As shown on Figure 2, the input sequence is processed by a BLSTM-CTC network in order to compte character posterior probabilities at every step. Then the probabilities of each labels are fed to the HMM stage (using class posteriors in place of the character likelihood computed by Gaussian Mixtures Models in the traditionnal HMM framework) to perform the alignment of the spotting model. We now describe the HMM line spotting models which enable us to spot either words (cf section 3.2) or REGEX (cf section 3.3).

3.2 Handwritten word spotting model

Our word spotting model describes a line of text that may contain the word to spot. As it is classically proposed in the literature, it is made of the HMM word model surrounded by filler models that represent any other sequence of characters. Figure 1 shows an example of a word spotting model for the word **”sentiments”**. The space model is directly integrated into the filler. By constraining the whole model, we can locate the word at the beginning, in the middle or at the end of the line. The filler model is basically an ergodic model made of every character model. In our problem, we use 99 models corresponding to lower and upper cases, digits, punctuations and space.

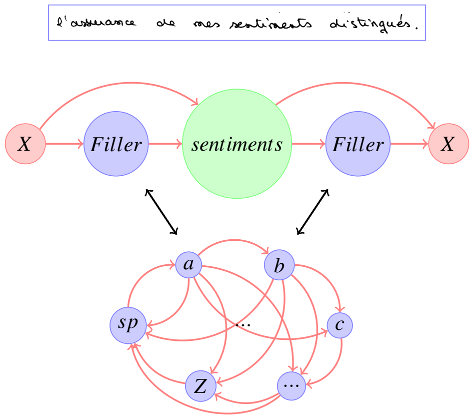


Figure 1: HMM line model : Detail of every component of the line model

Decoding a text line is classically achieved using the Viterbi algorithm, the system will outputs the character sequence with the maximum likelihood $P(X|\lambda)$. In order to accept the spotted word or reject it, decoding is generally performed twice: a first pass using the spotting model, and a second pass using a filler model. The likelihood ratio of the two models serves generally as a score for accepting or rejecting the spotted hypothesis. Using the BLSTM-CTC architecture, posterior probabilities are computed that can directly serve as a score for accepting/rejecting the hypothesis, without the need for a filler model. The score of each spotted hypothesis is computed by the average character posteriors over the number of frames spanning the hypothesis. This score is then normalised by the number of characters of the spotted word. Doing this, we choose to rely on the strong discriminative decisions of the BLSTM-CTC and use the HMM only as a sequence model constrained by high level information such as lexicons and/or language models. The graphical representation of the whole word spotting system is shown on Figure 2

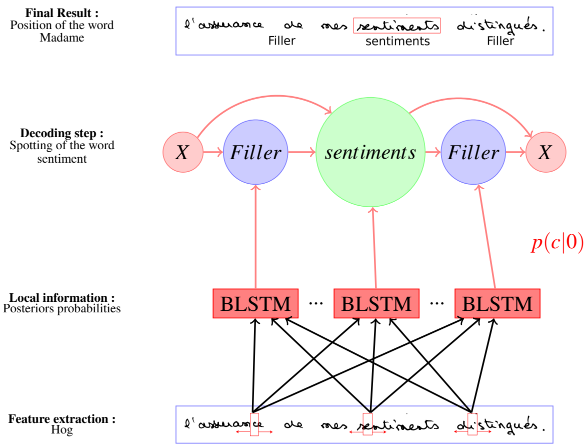


Figure 2: Hybrid structure BLSTM/HMM : Details of every step of the word spotting task from feature extraction to position of the word **sentiments** in the sentence. The BLSTM/CTC outputs a posteriori probabilities for HMM decoding.

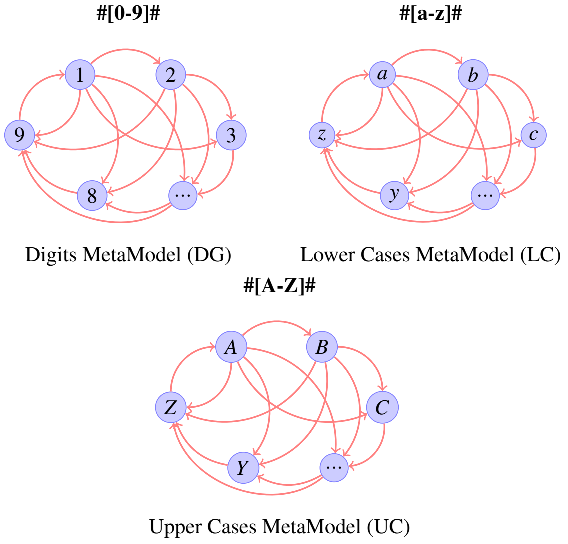
We now show how this model can be adapted to REGEX spotting.

3.3 Regular expression spotting model

As previously mentionned, REGEX spotting is a generalisation of the word spotting task, the difference is that the sequences to spot are less constrained and more variable, thus leading to a larger lexicon of admissible expressions.

In order to cope with REGEX queries, we use the HMM stage to model a regular expression with a stochastic model of character sequences. Each meta model is an ergodic model of characters implied in the query, e.g, Lower Cases (**#[a-z]#**), Upper Cases (**#[A-Z]#**) or Digits (**#[0-9]#**), as it is the case for the Filler models. Figure 3 shows examples of meta models for these three examples.

We also need to model the variable length of the queries, which may occurs when using ***** or **+** operators (spotting between 0 and ∞ times a character, or spotting between 1 and ∞ times a character) such as in **#[0-9]+#** which stands for any sequence of at least 1 digit. This is simply modeled by allowing auto transitions over the desired character meta model. Figure 4 shows an example of a model for spotting variable length sequences. The query taken is the sub-string **agr** following by an unconstrained sequence of lower cases (**#[a-z]*#**), in this example we hope that the system will spot the word **agréer** correctly.



Digits MetaModel (DG)

Lower Cases MetaModel (LC)

Upper Cases MetaModel (UC)

Figure 3: HMM MetaModels

The following models allow searching for a REGEX at the beginning of a line (**#[a-z]*ion#**), at the end of a line ((**#le[a-z]*#**)), or both (**#[A-Z]o[a-z]#**). The line model can also only contain meta models dedicated to spotting sequences of digits of any length, for example (**#[0-9]*#**) or word beginning by one upper case character and ending with a sequence of lower cases characters of arbitrary length (**#[A-Z][a-z]*#**). Here, the arbitrary length of the sequence unconstrained (*****) is controled by the auto-transition probabilities of the meta model of the HMM.

As the transitions in the HMM meta models are ergodic, the Viterbi alignment will only be driven by the local classification of BLSTM-CTC. The spotting model depends on its discriminant capacity to feed the higher HMM stage with accurate information from the local character recognition stage.

The graphical representation of the whole REGEX spotting system is shown on Figure 5.

Finally, the integration of meta models and auto transitions into the line model allows spotting of handwritten REGEX. Practically, the line model is build on the fly at the time of querying the data-set, by rewriting the REGEX into a HMM line spotting model. At this time, the ”translation” is manually done, but an automatization of this task can be performed for industrial purpose.

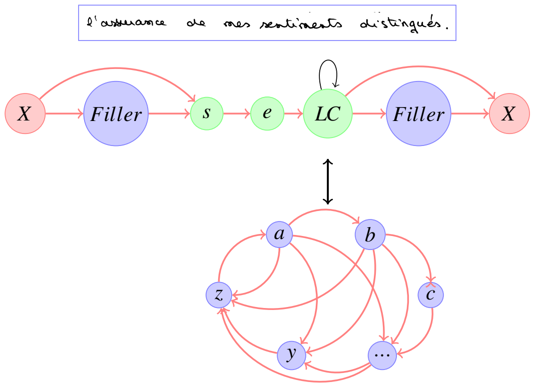


Figure 4: HMM stage : Spotting of regular expressions **#se[a-z]*#** (i.e every word beginning by the sub-string **se** followed by any number of lower case characters).

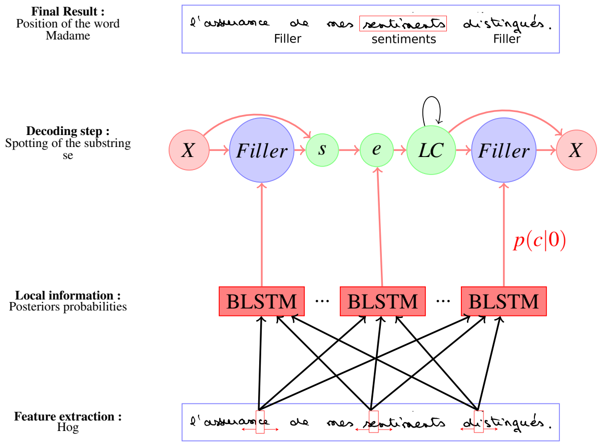


Figure 5: Hybrid structure BLSTM/HMM : Details of every step of the REGEX spotting task from feature extraction to position of the REGEX **#se[a-z]*#** in the sentence.

4 EXPERIMENTS

In this section, we give some details about the implementation of the system, starting with a description of the features extraction in section 4.1. The performance of the system are evaluated using the 2011 RIMES database(Grosicki and El-Abed, 2011), they are summarized in section 4.2.

4.1 Features Set

Our feature vector is based on Histograms of Oriented Gradient (HOG) (Rodrıguez and Perronnin, 2008) extracted from windows of 8×64 pixels. During the extraction, the window is dividing into sub-windows of $n \times m$ pixels. For each sub-window a histogram is computed, representing the distribution of the local intensity gradients (edge direction). The histograms of every windows are then merged to obtain the final representation of our feature vector representation. We used 8×8 non-overlapping sub-windows using 8 directions, this produces a 64 dimensional feature vector.

4.2 Results and discussion

To evaluate the performance of our system, all the experiments have been performed on the RIMES database used for the 2011 ICDAR handwriting recognition competitions (Grosicki and El-Abed, 2011). The training database is composed of 1.500 documents, the validation and test sets are composed respectively of 100 documents. In order to evaluate the spotting system, we compute recall (R) and precision measures (P). To do this, the number of true positives (TP), false positives (FP), and false negatives (FN) are evaluated for all possible threshold values. From these values, a recall-precision curve is computed by accumulating these values over all word queries.

$$R = \frac{TP}{TP+FN} \quad P = \frac{TP}{TP+FP} \tag{1}$$

4.2.1 Regular expression results

To evaluate the performance of our system on a regular expression spotting task we performed exactly the same experiments as in (Kessentini et al., 2013). In this study the authors were interested in spotting 4 different REGEX queries corresponding to the the search for the sub-strings ”effe”, ”pa”, ”com” and ”cha” at the beginning of a word (**#effe[a-z]*#**, **#pa[a-z]*#**, **#com[a-z]*#**, **#cha[a-z]*#**). As for word spotting experiments, results of the HMM system have been added too in order to provide a precise comparison between those systems (cf Figure 6-9).

A first observation is that the system achieves good performance, since most of the REGEX queries lead to a mean-average precision of nearly 75%, whereas the queries involve many fewer constraints than for word spotting. Moreover, our results are far beyond the standard HMM approach. We can observe a gap of more than 40% in the difficult cases (**#com[a-z]*#**) and (**#cha[a-z]*#**) and 20% in easier ones (**#effe[a-z]*#** and **#pa[a-z]*#**). We also run more test on other queries such as **#[a-z]*er#**,**#[a-z]*tion#**, **#[a-z]*tt[a-z]*#** and **#[a-z]*mm[a-z]*#**. The results are still pretty good for both textbf#[a-z]*tion# **#[a-z]*mm[a-z]*#**. However, the system seems to have trouble dealing with the two other queries. It is certainly due the fact that two consecutive letters such as

”t” are really difficult to spot. Concerning the other issue, it is due to the high level of confusion between ”r” and other characters such as ”n”,”u”, etc.

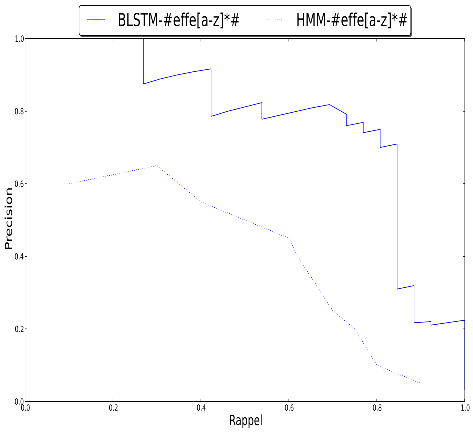


Figure 6: Regular expression spotting performance with the sub-string **effe** (**#effe[a-z]*#**)

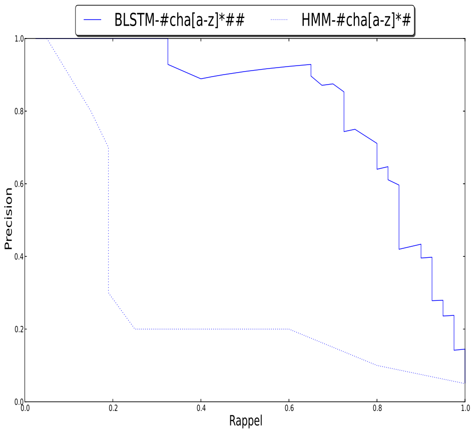


Figure 7: Regular expression spotting performance with the sub-string **cha** (**#cha[a-z]*#**)

We have also tested less constrained queries, with the search for REGEX containing any sequence of upper cases characters (**#[A-Z]*#**), and any sequence of digits (**#[0-9]#**). This problem is by far more difficult than the previous queries since the corresponding sequences may have variable contents and lengths. For example the digit query should detect the sequence *”1”* as well as sequence *”0123456789”*. Results are presented in Figure 11.

Knowing the difficulty of the problem, the performance are still interesting. Note that digit characters are not very frequent in the database. An interesting fact is that the Uppercase query can reach interesting precision scores, whereas the digit query can reach very high recall scores.

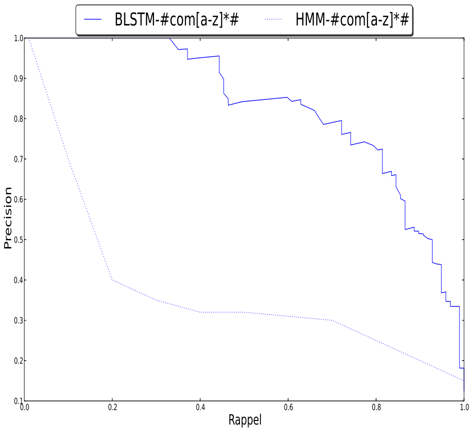


Figure 8: Regular expression spotting performance with the sub-string **com** (**#com[a-z]*#**)

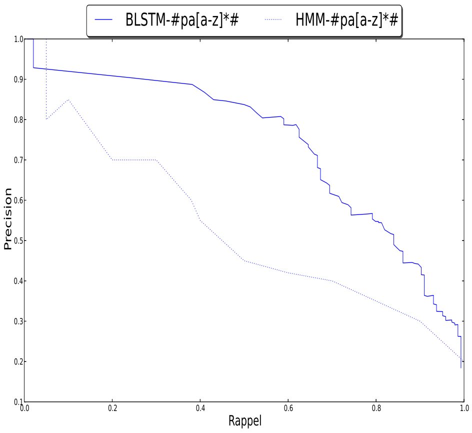


Figure 9: Regular expression spotting performance with the sub-string **pa** (**#pa[a-z]*#**)

5 CONCLUSION

In this paper, we have proposed a hybrid system BLSTM-CTC/HMM able to spot any word of REGEX. We have shown that the hybrid system exhibits interesting results, even on weakly constrained queries such as the search for sequences of digits of arbitrary length. We have compared our system for REGEX spotting with some recent work carried out on the same data-set and using the standard HMM framework. Our approach outperforms this system by more than 30% on the standard word spotting task and by more than 40% on REGEX spotting. These very promising results allow to envisage the application of higher level spotting systems such as addresses, named entities for which a combination of specific markers (keywords and alpha numerical expressions) is generally used to detect the relevant information.

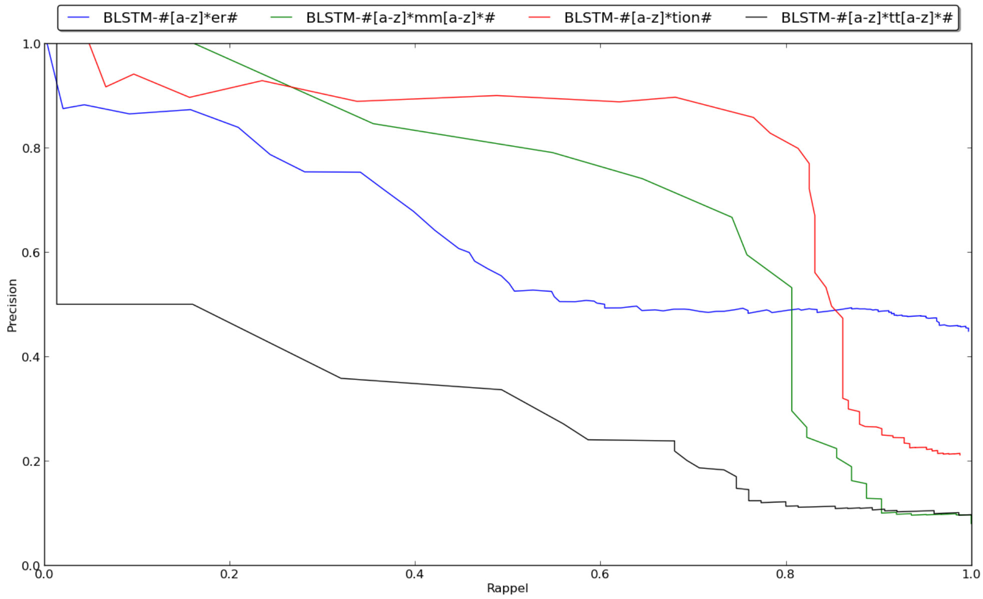


Figure 10: Regular expression spotting performance for **#[a-z]*er#**,**#[a-z]*mm[a-z]*#**, **#[a-z]*tion#** and **#[a-z]*tt[a-z]*#**

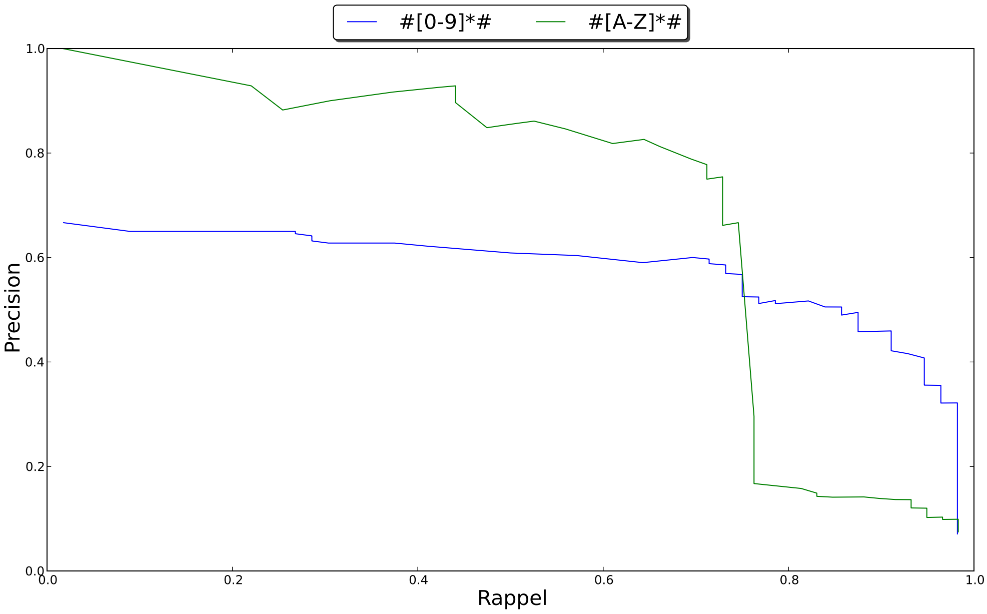


Figure 11: Regular expression spotting performance with upper case sequence (**#[A-Z]*#**) and number sequence (**#[0-9]#**)

REFERENCES

Adamek, T., OConnor, N. E., and Smeaton, A. F. (2007). Word matching using single closed contours for indexing handwritten historical documents. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(2-4):153–165.

Cao, H. and Govindaraju, V. (2007). Template-free word spotting in low-quality manuscripts. In *Proceedings of the 6th International Conference on Advances in Pattern Recognition*, pages 135–139.

Chatelain, C., Heutte, L., and Paquet, T. (2006). A two-stage outlier rejection strategy for numerical field extraction in handwritten documents. In *ICPR, Hong Kong, China*, volume 3, pages 224–227.

Chatelain, C., Heutte, L., and Paquet, T. (2008). Recognition-based vs syntax-directed models for numerical field extraction in handwritten documents. In *ICFHR, Montreal, Canada*, page 6p.

Dengel, A. R. and Klein, B. (2002). smartfix: A requirements-driven system for document analysis and understanding. In *Document Analysis Systems V*, pages 433–444. Springer.

Fischer, A., Keller, A., Frinken, V., and Bunke, H. (2012). Lexicon-free handwritten word spotting using character hmms. *Pattern Recognition Letters*, 33(7):934–942.

Frinken, V., Fischer, A., and Bunke, H. (2010). A novel word spotting algorithm using bidirectional long short-term memory neural networks. In Schwenker, F. and El Gayar, N., editors, *Artificial Neural Networks in Pattern Recognition*, volume 5998 of *Lecture Notes in Computer Science*, pages 185–196. Springer Berlin Heidelberg.

Frinken, V., Fischer, A., Manmatha, R., and Bunke, H. (2012). A novel word spotting method based on recurrent neural networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):211–224.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM.

Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J., and Fernández, S. (2008). Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 577–584.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):855–868.

Grosicki, E. and El Abed, H. (2009). Icdar 2009 handwriting recognition competition. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1398–1402. IEEE.

Grosicki, E. and El-Abed, H. (2011). Icdar 2011-french handwriting recognition competition. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1459–1463. IEEE.

Hosoya, H. and Pierce, B. (2001). Regular expression pattern matching for xml. In *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 67–80.

Kessentini, Y., Chatelain, C., and Paquet, T. (2013). Word spotting and regular expression detection in handwritten documents. In *ICDAR*.

Morita, M. E., Sabourin, R., Bortolozzi, F., and Suen, C. Y. (2003). Segmentation and recognition of handwritten dates: an hmm-mlp hybrid approach. pages 248–262.

Rath, T. M. and Manmatha, R. (2003). Features for word spotting in historical manuscripts. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 218–222. IEEE.

Rodrıguez, J. A. and Perronnin, F. (2008). Local gradient histogram features for word spotting in unconstrained handwritten documents. In *Int. Conf. on Frontiers in Handwriting Recognition*.

Rodríguez-Serrano, J. A. and Perronnin, F. (2009). Handwritten word-spotting using hidden markov models and universal vocabularies. *Pattern Recognition*, 42(9):2106–2116.

Rodríguez-Serrano, J. A., Perronnin, F., Lladós, J., and Sánchez, G. (2009). A similarity measure between vector sequences with application to handwritten word image retrieval. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1722–1729. IEEE.

Rusinol, M., Aldavert, D., Toledo, R., and Lladós, J. (2011). Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 63–67. IEEE.

Spitz, A. (1995). Using character shape codes for word spotting in document images. In *In: Proceedings of the symposium on document analysis and information retrieval*, pages 382–389.

.Spitz, A. (1997). Determination of script, language content of document images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 235–245.

Thomas, S., Chatelain, C., Heutte, L., and Paquet, T. (2010). An information extraction model for unconstrained handwritten documents. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3412–3415. IEEE.

Wöllmer, M., Eyben, F., Graves, A., Schuller, B., and Rigoll, G. (2009). A tandem blstm-dbn architecture for keyword spotting with enhanced context modeling. In *Proc. of NOLISP*.

Wshah, S., Kumar, G., and Govindaraju, V. (2012). Script independent word spotting in offline handwritten documents based on hidden markov models. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 14–19. IEEE.