Using top n Recognition Candidates to Categorize On-line Handwritten Documents

Sebastián Peña Saldarriaga
LINA - UMR CNRS 6241
Université de Nantes
sebastian.pena-saldarriaga@univ-nantes.fr

Emmanuel Morin
LINA - UMR CNRS 6241
Université de Nantes
emmanuel.morin@univ-nantes.fr

Christian Viard-Gaudin
IRCCyN - UMR CNRS 6597
École Polytechnique de l'Université de Nantes
christian.viard-gaudin@univ-nantes.fr

Abstract

The traditional weighting schemes used in text categorization for the vector space model (VSM) cannot exploit information intrinsic to texts obtained through on-line handwriting recognition or any OCR process. Especially, top n $(n > 1)$ *candidates could not be used without flooding the resulting text with false occurrences of spurious terms. In this paper, an improved weighting scheme for text categorization, that estimates the occurrences of terms from the posterior probabilities of the* top n *candidates, is proposed. The experimental results show that the categorization performances increase for texts with high error rates.*

1. Introduction

Digital pens, PDAs and pen-based devices are emerging technologies that are growing in importance. In recent years, the achievement in handwriting recognition has promoted on-line handwriting signals, produced with such devices, as a new source of documents in natural language. The data obtained through an on-line recognition process is *noisy*, i.e. it contains word insertions, deletions and substitutions referring to the text actually contained in the original signal. Noise presents a serious challenge in the downstream applications making use of recognized documents as input.

The interest for problems relating to processing text data from loosely constrained noisy sources has grown in the recent years [5]. Researchers have proposed different methods to deal with named entity recognition, parsing, indexation and retrieval of noisy texts for instance. Since printed documents can be easily collected, most of this research has so far been conducted with texts obtained through Optical Character Recognition (OCR).

While categorization of clean electronic texts has been thoroughly studied, the problem of noisy document categorization has little been addressed, and in particular for noisy documents coming from a handwritten source. Two works [13, 6] that addressed this matter showed that a performance gap can be observed when comparing the performances of a categorization system over texts obtained through handwriting recognition and the same texts available as ground truth. The significance of this gap depends on the recognized document quality, and the learning algorithm used. Both works use as categorization input the most likely word sequence given by a recognition system.

While the recognition system produces noise, it can also give hints about the quality of the recognized text. A confidence score can be given for each word, furthermore a list of n word recognition candidates can be obtained.

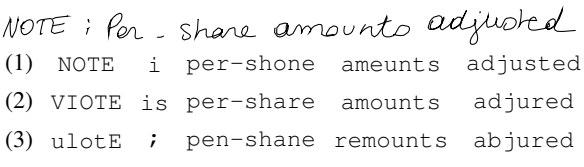

Figure 1. Recognition with top 3 candidates

The candidates are sorted by the confidence score or a posterior probability as shown in figure 1. Our hypothesis is that the use of this information might help fulfill the performance gap previously observed between electronic and

recognized texts.

The goal of this paper is to perform text categorization using the top n ($n > 1$) word recognition candidates rather than the standard approach usually limited to the top 1 candidate. But using many word candidates with equal weights will introduce spurious words, hence affecting categorization performances. This means that the standard weighting schemes used for the vector space model (VSM) [9] have to be modified in order to catch occurrences of relevant terms while minimizing the effects due to the introduction of false acceptances of words.

The remainder of this paper is organized as follows. The general categorization process and the modified weighting scheme are presented in section 2. In section 3, the recognition module that allows for the production of a transcribed version of handwritten document with several word recognition candidates is introduced . Results of document recognition and subsequent categorization are presented and discussed in section 4. Finally, section 5 draws conclusions and lines out future work.

2. Text categorization using top n candidates

In the VSM, and the categorization methods based on it, each occurrence of an indexation term is essential for prediction of document categories. However, when errors exist in a document obtained through on-line handwriting recognition, some terms might not be present due to an incorrect recognition.

We propose to use the top n recognition candidates rather than the typical output of the recognition system usually containing the top choice candidate. The use of top n candidates can help counting occurrences of missing terms, because the chance of the correct term being in the top n increases as n does. However, using too many candidates will introduce false occurrences of words, thus making the text noisier.

2.1. Text categorization

Text categorization is often performed by machine learning algorithms based on the VSM [10]. Two state-of-the-art categorization methods are used in our experiments: a k-nearest neighbour (kNN) algorithm and Support Vector Machines (SVM) [12].

Typically, before one of these machine learning algorithm can be used, two steps should be performed:

- **Text normalization**: First stopwords, which are assumed to carry no information, are removed. In order to reduce declension effects in text, stemming [8] or suffix stripping is performed on the remaining words.
- **Term selection and weighting**: Term selection is used to define the vector space. The goal of term selection is to choose the terms which are relevant to our categorization task. Then, since machine learning algorithms are based on the VSM, texts must be transformed into vectors. The resulting term list is weighted using an association measure such as the normalized $tf \times idf$ score [11].

Both of these operations are based on raw occurrences of terms. In order to avoid side effects due to false occurrences of spurious terms, careful work has to be done to estimate the frequency of the word recognition candidates.

2.2. A weighting scheme for top n candidates

Top n candidates are often ranked according to their posterior probability. This is true for some recognition systems, and in particular for the one we used in our experiments. In order to evaluate the importance of specific candidate-terms, we need to define the candidate-term frequency.

Definition 1 *Candidate-term frequency*

Let $\boldsymbol{p_n(i)}$ be the probability of the $\boldsymbol{n}$-th occurrence of the candidate-term $\boldsymbol{i}$, and $\boldsymbol{N}$ the occurrences of $\boldsymbol{i}$ in a recognized document $\boldsymbol{d}$. The frequency of the candidate-term $\boldsymbol{i}$ is defined as follows:

$$ctf(i) = \sum_{n=1}^{N} p_n(i) \tag{1}$$

In order to reduce, text-length effects, the $ctf(i)$ should be normalized.

Definition 2 *Normalized candidate-term frequency*

Let $\boldsymbol{M}$ be the number of indexation terms, and $\boldsymbol{i}$ a given candidate-term. The normalized candidate-term frequency of $\boldsymbol{i}$ is defined as follows:

$$nctf(i) = \frac{ctf(i)}{\sum_{j=1}^{M} ctf(j)} \tag{2}$$

A normalized $tf \times idf$ score for candidate-terms in the output of a recognition system can be computed using the ordinary idf and the nf_{ct_i} score.

Definition 3 *Normalized candidate-tf $\times$ idf*

Let $\boldsymbol{K}$ be the number of documents in a collection, and $\boldsymbol{k_i}$ the number of documents in the collection containing the candidate-term $\boldsymbol{i}$. The weight of the candidate-term $\boldsymbol{i}$ in a vector is defined as follows:

$$ctf.idf(i) = \frac{ctf(i) \times \log \frac{K}{k_i}}{\sqrt{\sum_{j=1}^{M} (ctf(j) \times \log \frac{K}{k_j})^2}} \tag{3}$$

The definition of this new weighting scheme for candidate terms allows us to apply standard categorization methods to our recognized data, without major modifications.

Whenever the posterior probability of a term t is not given by the recognizer, a probability can be estimated according to the rank of t within the top n candidate list [3].

2.3. Performance evaluation

Several measures exist to evaluate the quality of categorization. The most commonly used effectiveness measures are recall and precision. As we aim for an application where documents are available at different moments in time; the system evaluation should be done on a document basis, i.e using micro-averaged precision and recall [1]. Moreover, as we work with single label documents; micro-averaged recall and precision are equal. Hence, a single accuracy measure will be given as system evaluation: the classification rate R. With respect to the category associated to each document, the classification rate is defined as follows.

$$R = \frac{d}{D} \tag{4}$$

Where d is the number of documents correctly assigned to a category, and D the number of documents in the collection we intend to categorize.

3. On-line Handwriting Recognition

The recognition, that enables the production of text from an on-line handwritten document, is performed using the recognition engine of MyScript Builder [1]. The recognition module allows the use of linguistic knowledge, two different resources are used:

lk-text is composed of a standard lexicon of English words and a statistical language model. This model helps detect the most likely sequence of words, for example, *'this is'* will have priority over *'this in'*.

lk-free helps detect the most likely sequence of characters. When the recognizer is unable to differentiate between an *'I'* and a *'1'*, if the other characters in the word are recognized as uppercase letters then *'I'* will have priority over *'1'*.

The recognition system is evaluated using the Word Error Rate (WER) and the Term Error Rate (TER). The WER is a common metric of performance used in OCR. It is the proportion of words incorrectly recognized from the original text.

[1]MyScript Builder SDK can be found at *http://www.visionobjects.com/products/software-development-kits/myscript-builder*

$$WER = 1 - \frac{\sum_i^W \min\left(wf(i), wf'(i)\right)}{\sum_k^W wf(k)} \tag{5}$$

Where $wf(i)$ and $wf'(i)$ are the frequencies of the word i in the clean and recognized texts respectively, and W the number of words in text.

The TER [13] is a judicious measure for downstream applications that use text normalization. The TER is calculated by:

$$TER = 1 - \frac{\sum_i^T \min\left(tf(i), tf'(i)\right)}{\sum_k^T tf(k)} \tag{6}$$

Where $tf(i)$ and $tf'(i)$ are the frequencies of the term i in the clean and recognized texts respectively, and T the number of terms in the text.

4. Experiments

This section reports experiments performed using an online handwritten corpus. It consists of 1,625 documents for training and 404 for testing purposes. These documents are a subset of the Reuters-21578 corpus distributed among 10 categories. An example document from the handwritten corpus is shown in figure 2.

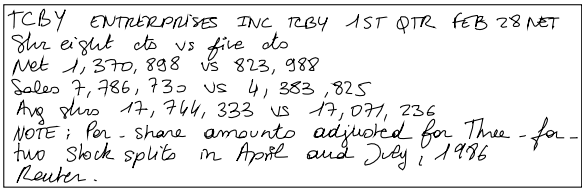

Figure 2. Handwritten document

4.1. Recognition

All the documents, in the training and test sets, are recognized using MyScriptBuilder with both resources, *lk-text* and *lk-free*. Since the recognition is consistent and behaves very similarly regard to both the training and test sets, only the errors rates of the test set documents are reported below.

Figure 3 shows the evolution of the WER and TER according to the number of recognition candidates accepted.

The recognition performances of *lk-text* clearly outperform *lk-free* results. The performances of a recognition system can be improved by incorporating statistical information at the word-sequence level [7], i.e. a language model. In the absence of such knowledge, which is the case for *lk-free*, the output of a recognition system is known to be very noisy.

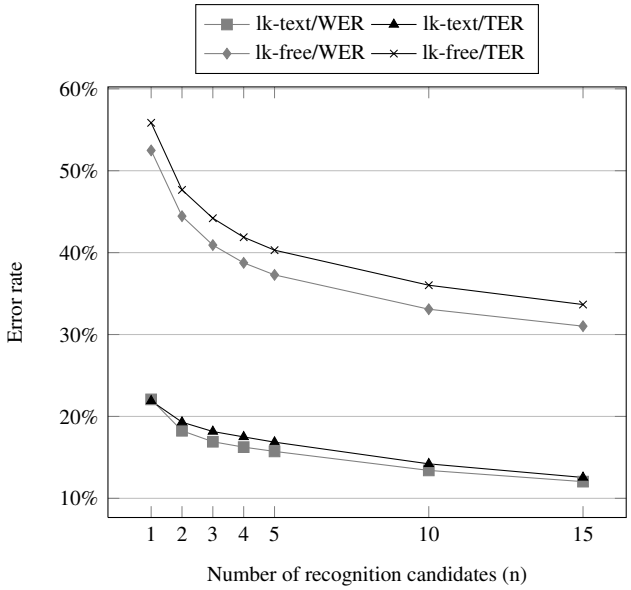

Figure 3. Recognition error rates

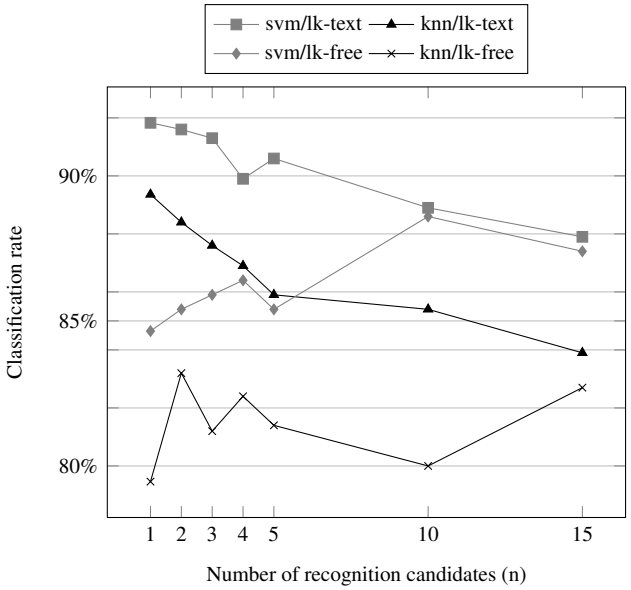

Figure 4. Classification rates

Unsurprisingly, both error rates decrease as the number of recognition candidates increases because the probability of the correct word or term being in the candidate list increases according to the list's size.

4.2. Categorization

Categorization of the 404 test documents was performed on both, typical recognition output (containing the top candidate) and the recognition output containing several candidates. Two categorization methods have been tested: a kNN algorithm and Support Vector Machines [4]. It is worthwhile to note that the handwritten documents used for training the categorization algorithms are not used for training the recognizer, which is a stable and ready-to-use tool. The training consists mainly in the selection of the relevant terms defining the VSM.

The parameters of the categorization classifiers have been tuned to achieve maximum accuracy. This was done using a subset of the single top-candidate recognition output, as validation set. The optimal parameters for the kNN algorithm are $k = 15$ neighbours and 300 relevant terms, while 1,000 relevant terms are optimal for the SVM classifier. The term selection has been performed using George Forman's round robin algorithm [2] over category specific scores obtained with the χ^2 statistic [14].

Figure 4 shows the categorization rates following the categorization algorithm, the recognition resource, and the number of candidates used.

The performances obtained with *lk-text* and the top-candidate output are better than those obtained with a candidate list. The accuracy cannot be improved this way, on the contrary it decreases as the candidate list size increases. The top-candidate recognition is already good enough for categorization with a reasonably good classification rate.

On the other hand, using the candidate list with *lk-free* helps improve categorization accuracy. As a matter of fact, with recognition outputs for all $n > 1$, both classifiers yields better results than the top-candidate output. However, the accuracy gain is not regular and does not seem correlated with n. For SVM, the mean categorization rate increase is 1.87% with a standard deviation of 1.27%. Whereas for kNN the mean increase is 2.36% with a standard deviation of 1.17%.

5. Conclusions

Noise is pervasive in applications processing texts obtained through recognition of any signal intended for human communication such as speech or handwriting. Despite the growing interest for researchers on the effects on noise in downstream applications, some areas remain unexplored. In particular, categorization of noisy texts coming from a handwritten source has little been studied. Error rates in this kind of documents are often substantially higher than for machine printed text recognition.

The effects of noise in the categorization of handwriting recognition outputs has been recently studied [13, 6]. Nevertheless, authors do not make use of some information coming along with the recognized texts: the top n

word recognition candidates and the associated confidence scores.

In this paper we proposed a modified $tf \times idf$ measure for estimating the weight of terms, from the posterior probabilities of the top n recognition candidates, in a vector space. Experiments were performed using an on-line handwritten version of a Reuters-21578 corpus subset. Different linguistic resources are used within the recognition engine in order to obtain different levels of noise. First, categorization is performed on the usual output given by the recognition system (containing the top-choice candidate) using the standard $tf \times idf$ weighting, then the process is repeated over all the other versions containing up to 15 candidates with the improved weighting scheme.

On the one hand, categorization results on the *lk-text* set show that no performance improvement should be expected for $n > 1$. Texts with WER and TER below 22% are better categorized using the top-choice recognition. On the other hand, an accuracy improvement of more than 3% can be observed for poorly recognized texts where TER and WER stand above 50%. However, this improvement is not regular and cannot be related to the number of candidates used (n) in any way. The major improvement obtained with SVM uses $n = 10$ while kNN performs better with $n = 2$.

When used with *lk-text*, the new weighting scheme was not sufficient to prevent spurious terms from affecting categorization performances. A thresholding strategy for recognition candidates might help reducing side effects due to very unlikely candidates, and will be explored in futur work.

The overall results presented in this work confirm our feeling that the relationship between the noise which is present in the output of an on-line handwriting recognition system and its effects on the categorization is not well understood. While waiting for new advances in handwriting recognition that allow for the production of better, less noisy, documents, future work should focus on this matter.

Acknowledgments

This work is funded by La Région Pays de la Loire under the MILES Project and by The French National Research Agency under the framework of the program Technologies Logicielles (ANR-06-TLOG-009). The authors are indebted with the anonymous writers who willingly helped them to the tedious task of rewriting economic newswires.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*, chapter Retrieval Evaluation, pages 73–99. Addison-Wesley, 1999.

[2] G. Forman. A pitfall and solution in multi-class feature selection for text classification. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML '04)*, 2004.

[3] N. R. Howe, T. M. Rath, and R. Manmatha. Boosted decision trees for word recognition in handwritten document retrieval. In *Proceedings of the Twenty-Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, pages 377–383, 2005.

[4] T. Joachims. *Learning to Classify Text using Support Vector Machines*. Kluwer Academic Publishers, 2002.

[5] D. Lopresti, S. Roy, K. Schulz, and L. V. Subramaniam, editors. *Proceedings of SIGIR 2008 Workshop on Analytics for Noisy Unstructured Text Data*, 2008.

[6] S. Peña Saldarriaga, C. Viard-Gaudin, and E. Morin. On-line handwritten text categorization. In *Document Recognition and Retrieval XVI, IS&T/SPIE International Symposium on Electronic Imaging (DRR 2009)*, page 724709, 2009.

[7] F. Perraud, C. Viard-Gaudin, E. Morin, and P. M. Lallican. Statistical language models for on-line handwriting recognition. *IEICE Transactions on Information and Systems, Special Issue on Document Image Understanding and Digital Document*, E88-D(8):1807–1814, 2005.

[8] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[9] G. Salton, A. Wong, and C. S. Wang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[10] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[11] K. Spärck Jones. Experiments in relevance weighting of search terms. *Information Processing and Management*, 15:133–144, 1979.

[12] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[13] A. Vinciarelli. Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1882–1895, 2005.

[14] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, pages 412–420, 1997.