On Authorship Attribution via Markov Chains and Sequence Kernels

Conrad Sanderson and Simon Guenter
Australian National University, Canberra, ACT 0200, Australia
National ICT Australia (NICTA[*]), Locked Bag 8001, ACT 2601, Australia
conradsand @ ieee.org, simon.guenter @ rsise.anu.edu.au

Abstract

We investigate the use of recently proposed character and word sequence kernels for the task of authorship attribution and compare their performance with two probabilistic approaches based on Markov chains of characters and words. Several configurations of the sequence kernels are studied using a relatively large dataset, where each author covered several topics. Utilising Moffat smoothing, the two probabilistic approaches obtain similar performance, which in turn is comparable to that of character sequence kernels and is better than that of word sequence kernels. The results further suggest that when using a realistic setup that takes into account the case of texts which are not written by any hypothesised authors, about 5000 reference words are required to obtain good discrimination performance.

1. Introduction

The task of authorship attribution is to deduce the author of a given text (e.g. e-mails, articles, books). Applications include attribution of threatening or inappropriate messages sent anonymously or under a pseudonym, plagiarism detection, as well as resolving questions of disputed authorship. One example is the forensic analysis of the Unabomber manifesto [6].

Within the realm of automatic author attribution, recently it has been shown that encouraging performance can be achieved via the use of probabilistic models based on n-grams [3] and Markov chains of characters and words [9]. Separately, Support Vector Machines (SVMs), using the bag-of-words kernel, have been shown to obtain promising performance [5], while in another study, SVMs with kernels based on character collocations obtained mixed performance [4].

The abovementioned studies have several limitations. In [3], only a rudimentary probability smoothing technique was utilised to handle n-grams which were unseen during the training phase. In the dataset used by [9], each author tended to stick to one or two topics, raising the possibility that the discrimination was based on topic rather than by author style. In [4, 9], the datasets were rather small (in [4] the largest contains texts from five authors, while in [9] from ten), indicating the results may not be generalisable. In [3, 9], the attribution of a given document was forced to one of the authors from a set of possible authors (i.e. a closed set identification setup), thus not taking into account the realistic case of a text which was not written by any of the authors. Lastly, all of the studies used different datasets and experiment setups, thus making a quantitative performance comparison of the different approaches infeasible.

Recently, various practical character and word sequence kernels have been proposed for text and biological sequence analysis [1, 7, 12]. This allows the possibility of kernel based techniques, such as SVMs, to be used in lieu of traditional probabilistic methods. In comparison to the latter, SVMs have the advantage of directly optimising the discrimination criterion.

This paper has two main aims: **(i)** to evaluate the usefulness of sequence kernel based approaches for the task of authorship attribution; **(ii)** to compare their performance with two probabilistic approaches based on Markov chains of characters and words. Several configurations of the sequence kernels are studied. The evaluations are done on a relatively large dataset (50 authors) where each author covers several topics. Moreover, rather than using a closed set identification setup, the evaluations are done using a verification setup. Here, a given text material is classified as either having been written by a hypothesised author or as not written by that author (i.e. a two class discrimination task).

The balance of this paper is structured as follows. Section 2 describes author attribution systems based on Markov chains of characters and words, while Section 3 describes the corresponding sequence kernel based approaches. Section 4 provides an empirical performance comparison of the abovementioned approaches. Section 5 concludes the paper by presenting the main findings and suggesting future work.

2. Markov Chain Based Approaches

The opinion on how likely a given text X was written by author A, rather than any other author, can be found by a log likelihood ratio:

$$\mathcal{O}_{A,G}\left(X\right) = |e_z(X)|^{-1} \log \left[\, p_A\left(e_z(X)\right) \,/\, p_G\left(e_z(X)\right) \right]$$

where $z \in \{words, chars\}$, $e_z(X)$ extracts an ordered set of items from X (where the items are either words or char-

[*] NICTA is funded by the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council. The authors are with the Statistical Machine Learning (SML) Program.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

acters, indicated by z), $|e_z(X)|^{-1}$ is used as a normalisation for varying number of items, while $p_A(e_z(X))$ and $p_G(e_z(X))$ estimate the likelihood of the text having been written by author A and a *generic* author[1], G, respectively.

Given a threshold t, text X is classified as having been written by author A when $\mathcal{O}_{A,G}(X) > t$, or as written by someone else when $\mathcal{O}_{A,G}(X) \leq t$. The $|e_z(X)|^{-1}$ normalisation term allows for the use of a common threshold (i.e. shared by all authors), which facilitates the interpretation of performance (e.g. via the use of the Equal Error Rate point on a Receiver Operating Characteristic curve [8]).

Appropriating a technique originally used in language modelling [2], the likelihood of author A having written a particular sequence of items, $X = \left(i_1, i_2, \cdots, i_{|X|}\right)$, can be approximated using the joint probability of all present m-th order Markov chains:

$$p_A(X) \approx \prod_{j=(m+1)}^{|X|} p_A\left(i_j | i_{j-m}^{j-1}\right) \tag{1}$$

where i_{j-m}^{j-1} is a shorthand for $i_{j-m} \cdots i_{j-1}$ and m indicates the length of the history. Given training material for author A, denoted as X_A, the maximum likelihood (ML) probability estimate for a particular m-th order Markov chain is:

$$p_A^{ml}\left(i_j | i_{j-m}^{j-1}\right) = \mathcal{C}\left(i_{j-m}^{j} | X_A\right) / \mathcal{C}\left(i_{j-m}^{j-1} | X_A\right) \tag{2}$$

where $\mathcal{C}\left(i_{j-m}^{j} | X_A\right)$ is the number of times the sequence i_{j-m}^{j} occurs in X_A. For chains that have not been seen during training, elaborate smoothing techniques [2] are utilised to avoid zero probabilities in Eqn. (1).

In this work we utilise interpolated Moffat smoothing[2], where the probability of an m-th order chain is a linear interpolation of its ML estimate and the smoothed probability estimate of the corresponding (m-1)-th order chain:

$$p_A^{mof}\left(i_j | i_{j\text{-}m}^{j\text{-}1}\right) = \left[1\text{-}\beta_{i_{j\text{-}m}^{j\text{-}1}}\right] p_A^{ml}\left(i_j | i_{j\text{-}m}^{j\text{-}1}\right) + \beta_{i_{j\text{-}m}^{j\text{-}1}}\, p_A^{mof}\left(i_j | i_{j\text{-}(m\text{-}1)}^{j\text{-}1}\right)$$

with the definition of β elucidated in [2]. The (m-1)-th order probability will typically correlate with the m-th order probability and has the advantage of being estimated from a larger number of examples. The 0-th order probability is interpolated with the uniform distribution, given by: $p_A^{unif} = 1 / |V_A|$, where $|V_A|$ is the vocabulary size [2].

When an m-th order chain has a history (i.e. the items i_{j-m}^{j-1}) which hasn't been observed during training, a back-off to the corresponding reduced order chain[3] is done:

$$\text{if } \mathcal{C}\left(i_{j-m}^{j-1} | X_A\right) = 0, \;\; p_A^{mof}\left(i_j | i_{j-m}^{j-1}\right) = p_A^{mof}\left(i_j | i_{j-(m-1)}^{j-1}\right)$$

Note that if the 0-th order chain also hasn't been observed during training, we are effectively backing off to the uniform distribution.

A caveat: the training dataset for an author can be much smaller (and hence have a smaller vocabulary) than the combined training dataset for the generic author, resulting in $p_A^{unif} > p_G^{unif}$. Thus when a previously unseen chain is encountered there can be a dangerous bias towards author A, i.e., $p_A^{mof}\left(i_j | i_{j-m}^{j-1}\right) > p_G^{mof}\left(i_j | i_{j-m}^{j-1}\right)$. To avoid this, p_A^{unif} must be set equal to p_G^{unif}.

3. Sequence Kernel Based Approaches

Kernel based techniques, such as SVMs, allow the comparison of, and discrimination between, vectorial as well as non-vectorial objects. In a binary SVM, the opinion on whether object X belongs to class -1 or +1 is given by:

$$\mathcal{O}_{+1,-1}(X) = \sum\nolimits_{j=1}^{|S|} \lambda_j \, y_j \, k(s_j, X) + b \tag{3}$$

where $k(X_A, X_B)$ is a symmetric kernel function which reflects the degree of similarity between objects X_A and X_B, while $S = (s_j)_{j=1}^{|S|}$ is a set of support objects with corresponding class labels $(y_j \in \{-1, +1\})_{j=1}^{|S|}$ and weights $\Lambda = (\lambda_j)_{j=1}^{|S|}$. The kernel function, b as well as sets S and Λ define a hyperplane which separates the +1 and -1 classes. Given a training dataset, quadratic programming based optimisation is used to maximise the separation margin [11].

Recently, kernels for measuring the similarity of texts based on sequences of characters and words have been proposed [1, 7, 12]. One kernel belonging to this family is:

$$k(X_A, X_B) = \sum\nolimits_{q \in Q^\star} w_q \, \mathcal{C}(q | X_A) \, \mathcal{C}(q | X_B) \tag{4}$$

where $Q^\star$ represents all possible sequences, in X_A and X_B, of the symbols in Q. In turn, Q is a set of possible symbols, which can be characters, e.g. Q = { 'a', 'b', 'c', $\cdots$ }, or words, e.g. Q = {'dog', 'cat', 'mouse', $\cdots$ }. Furthermore, $\mathcal{C}(q|X)$ is the number of occurrences of sequence q in X, and w_q is the weight for sequence q. If the sequences are restricted to have only one item, Eqn. (4) for the case of words is in effect a bag-of-words kernel [1].

In this work we have utilised weights that are dependent only on the length of each sequence, i.e. $w_q = w_{|q|}$. By default $w_{|q|} = 0$, modified by one of the following functions:

$$\begin{aligned}
\text{specific length: } & w_{|q|} = 1, \text{ if } |q| = \tau \\
\text{bounded range: } & w_{|q|} = 1, \text{ if } |q| \in [1, \tau] \\
\text{bounded linear decay: } & w_{|q|} = 1 + \tfrac{1-|q|}{\tau}, \text{ if } |q| \in [1, \tau] \\
\text{bounded linear growth: } & w_{|q|} = |q| \,/\, \tau, \text{ if } |q| \in [1, \tau]
\end{aligned}$$

where τ indicates a user defined maximum sequence length.

To allow comparison of texts with different lengths, a normalised version [11] of the kernel can be used:

$$\widehat{k}(X_A, X_B) = k(X_A, X_B) \,/\, \sqrt{k(X_A, X_A)\, k(X_B, X_B)}$$

with constraints $|X_A| \geq 1$ and $|X_B| \geq 1$.

It has been suggested that SVM discrimination based on character sequence kernels in effect utilises a noisy version of stemming [1]. As such, word sequence kernels could be more effective than character sequence kernels, since proper word stems, instead of full words, can be explicitly used. However, it must be noted that Eqn. (4) implicitly maps texts to a feature space which has one dimension for each of the possible sequences comprised of the symbols from Q [1]. When using words, the number of unique symbols (i.e. $|Q|$) is much greater than when using characters (e.g. 50,000 vs 128); furthermore, for a given text the number of words is always smaller than the number of characters. These observations indicate that for word sequence kernels the implicit feature space representation is of considerably higher dimensionality and can be sparser than for character sequence kernels, which could lead to poorer generalisation of the resulting classifier.

[1] A *generic* author is a composite of a number of authors.

[2] Moffat smoothing is often mistakenly referred to as Witten-Bell smoothing. See note 8 in [9] for an explanation.

[3] Personal correspondence with authors of [2].

No. characters	1750	3500	7000	14000	28000
No. words	312	625	1250	2500	5000

Table 1. Approximate correspondence between the number of characters and number of words.

4. Experiments and Discussion

We have compiled a dataset that is comprised of texts from 50 newspaper journalists, with a minimum of 10,000 words per journalist. Journalists were selected based on their coverage of several topics; any journalist who covered only one specific area (e.g. sports or economics) was not included in the dataset.

The experiments followed a verification setup, where a given text material was classified as either having been written by a hypothesised author or as not written by that author. The presentation of an *impostor text* (a text known not to be written by the hypothesised author) will be referred to as an *impostor claim*, while the presentation of a *true text* (a text known to be written by the hypothesised author) will be referred to as a *true claim*.

For a given text, one of the following two classification errors can occur: **(i)** a false positive, where an impostor text is incorrectly classified as a true text; **(ii)** a false negative, where a true text is incorrectly classified as an impostor text. The errors are measured in terms of the false positive rate (FPR) and the false negative rate (FNR). Following the approach often used within the biometrics field, the decision threshold was then adjusted so that the FPR is equal to the FNR, giving Equal Error Rate (EER) performance [8, 10].

The authors in the database were randomly assigned into two disjoint sections: **(i)** 10 background authors; **(ii)** 40 evaluation authors. For the case of Markov chain approaches, texts from the background authors were used to construct the generic author model, while for kernel based approaches they were used to represent the negative class. In both cases, text materials each comprised of approx. 28,000 characters were used, via randomly choosing a sufficient number of sentences from the pooled texts. Table 1 shows a correspondence between the number of characters and words, using the average word length of 5.6 characters including a trailing whitespace (found on the whole dataset).

For each author in the evaluation section, their material was randomly split into two continuous parts: training and testing. The split occurred without breaking sentences. The training material was used to construct the author model, while the test material was used to simulate a true claim as well as impostor claims against all other authors' models. Based on preliminary experiments, SVMs were trained by parting the training material into chunks; each of the chunks can become a support chunk. The minimum chunk size was set to 500 characters for the character kernels and to 4000 characters for the word kernels. Furthermore, stemming was used for the word based approaches (i.e. each word was replaced by its stem).

For each configuration of an approach (where, for example, the configuration is the order of the Markov chains), the above procedure was repeated ten times, with the randomised assignments and splitting being done each time. The final results were then obtained in terms of the mean and the standard deviation of the ten EERs (the standard deviations are shown as error bars in the result figures).

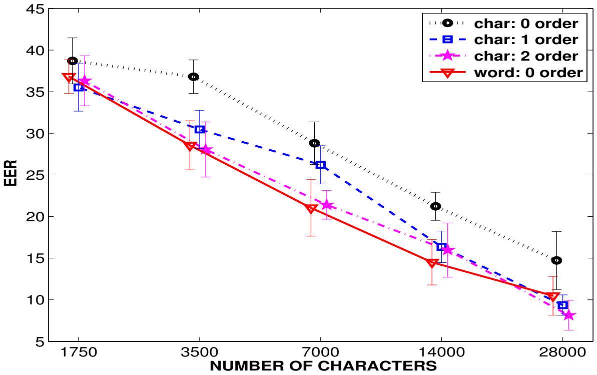

Figure 1. Performance of character and word Markov chain approaches.

In the first experiment we studied the effects of varying the order for character and word Markov chain approaches, while the amount of training *and* test material was decreased from approx. 28,000 to 1,750 characters (without breaking sentences).

Results in Fig. 1 show that the best performing word chain approach has an order of zero. Its performance is largely similar to the 2nd order character chain approach, with the latter obtaining a somewhat lower error rate at 28,000 characters. Higher orders of the word approach (not shown) have virtually the same performance as the 0th order. For the character approach, the difference in performance between 1st order and 2nd order chains could be considered as statistically insignificant due to the large overlap of the error bars.

In the second experiment we studied the effects of various weight functions and sequence lengths for the character sequence kernel. The amount of training and test material was fixed at approx. 28,000 characters. Results for specific length (Fig. 2) suggest that most of the reliable discriminatory information is contained in sequences of length 2. The error rates for the bounded range and bounded linear decay functions are quite similar, with both reaching minima for sequences of length 4; most of the improvement occurs when the sequences reach a length of 3. Hence while sequences with a specific length of 3 and 4 are less reliable than sequences with a specific length of 2, they contain (partly) complementary information which is useful when combined with information from shorter lengths. Emphasising longer lengths of 5 and 6 (via the bounded linear growth function) achieves a minor, but noticeable, performance degradation. We conjecture that the degradation is caused by the sparsity of relatively long sequences, which affects the generalisation of the classifier.

In the third experiment we compared the performance of character sequence kernels (using the bounded range function with τ=4) and several configurations of the word sequence kernels. The amount of training and test mate-

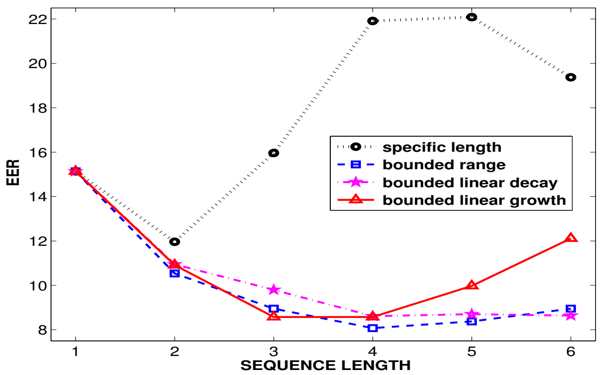

Figure 2. **Performance of the character sequence kernel approach for various weight functions.**

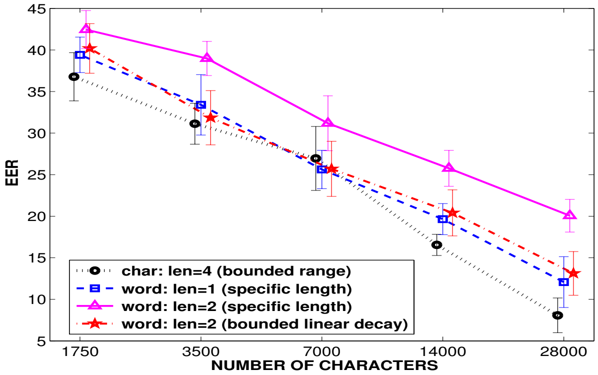

Figure 3. **Performance of character and word sequence kernel approaches.**

rial was decreased from approx. 28,000 to 1,750 characters. Fig. 3 shows that word sequences of length 2 lead to considerably worse performance than sequences of length 1 (i.e. individual words). Furthermore, the best performing combination of lengths (i.e. via the bounded linear decay function) does not provide better performance than using individual words. The character sequence kernel generally achieves a lower error rate than the best performing word sequence kernel. This suggests that the sparse feature space representation, described in Section 3, is becoming an issue.

By comparing Figs. 1 and 3 it can be observed that the best performing Markov chain based approaches (2nd order for characters and 0th order for words) obtain comparable performance to the character sequence kernel based approach (using the bounded range function with τ=4). The latter, in turn, has generally better performance than the best performing word sequence kernel based approach, which is in effect the bag-of-words approach.

5. Main Findings and Future Work

In this paper we investigated the use of character and word sequence kernels for the task of authorship attribution. We compared their performance with two probabilistic approaches based on Markov chains of characters and words. The evaluations were done on a relatively large dataset (50 authors), where each author covered several topics. Rather than using the restrictive closed set identification setup, a verification setup was used which takes into account the realistic case of texts which are not written by any hypothesised authors.

In the framework of Support Vector Machines, several configurations of the sequence kernels were studied, showing that word sequence kernels do not achieve better performance than a bag-of-words kernel. Character sequence kernels (using sequences with a length of 4) generally have better performance than the bag-of-words kernel and also have comparable performance to the two probabilistic approaches.

Interestingly, the bag-of-words kernel based approach obtains worse performance than the corresponding word based Markov chain approach. Apart from the issue of sparse feature space representation, factors such as the chunk size can also affect the generalisation performance.

When using the same number of words for both training and testing, the results also show that about 5000 words are required to obtain good discrimination performance. In future work it would be useful to evaluate the effects of training with a fixed number of words and testing with a varying number of words. Furthermore, more sophisticated character sequence kernels can be evaluated, such as mismatch string kernels used in bioinformatics, where mutations in the sequences are allowed [7].

References

[1] N. Cancedda, E. Gaussier, C. Goutte, J.-M. Renders. Word-Sequence Kernels. *J. Machine Learning Research*, 3:1059–1082, 2003.

[2] S.F. Chen, J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13:359–394, 1999.

[3] R. Clement, D. Sharp. Ngram and Bayesian Classification of Documents for Topic and Authorship. *Literary and Linguistic Computing*, 18(4):423–447, 2003.

[4] M. W. Corney. *Analysing E-Mail Text Authorship for Forensic Purposes*. Masters Thesis, Queensland University of Technology, Australia, 2003.

[5] J. Diederich, J. Kindermann, E. Leopold, G. Paass. Authorship Attribution with Support Vector Machines. *Applied Intelligence*, 19(1-2):109–123, 2003.

[6] D. W. Foster. *Author Unknown: On the Trail of Anonymous*. Henry Holt & Company, 2nd ed., 2001.

[7] C. Leslie, E. Eskin, A. Cohen, J. Weston, W. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004.

[8] J. Ortega-Garcia, J. Bigun, D. Reynolds, J. Gonzalez-Rodriguez. Authentication gets personal with biometrics. *IEEE Signal Processing Magazine*, 21(2):50–62, 2004.

[9] F. Peng, D. Schuurmans, S. Wang. Augmenting Naive Bayes Classifiers with Statistical Language Models. *Information Retrieval*, 7:317–345, 2004.

[10] C. Sanderson, S. Bengio, Y. Gao. On transforming statistical models for non-frontal face verification. *Pattern Recognition*, 39(2):288–302, 2006.

[11] B. Schölkopf, A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, USA, 2002.

[12] S.V.N. Viswanathan, A. Smola. Fast Kernels for String and Tree Matching. *Advances in Neural Information Processing Systems 15*, MIT Press, Cambridge, 2003, pp. 569–576.