Authorship Identification for Heterogeneous Documents

Yuta Tsuboi Yuji Matsumoto

Graduate School of Information Science, Nara Institute Science and Technology
8916-5 Takayama, Ikoma Nara 630-0101 Japan
{yuuta-t,matsu}@is.aist-nara.ac.jp

The study of authorship identification in Japanese has for the most part been restricted to literary texts using basic statistical methods. In the present study, authors of mailing list messages are identified using a machine learning technique (Support Vector Machines). In addition, the classifier trained on the mailing list data is applied to identify the author of Web documents in order to investigate performance in authorship identification for more heterogeneous documents. Experimental results show better identification performance when we use the features of not only conventional word N-gram information but also of frequent sequential patterns extracted by a data mining technique (PrefixSpan).

Keywords : Authorship Identification, E-mails, Web documents, Support Vector Machines, Sequential Pattern Mining, PrefixSpan

0[Jk%?%$%w$N%I%-%e%a%s%HKBP9$kCx<T?dDj

DZ0fM4B@ >>K\M5<#

F‘NI@hC<2J3X5;=QBg3X1!Bg3X>pJs2J3X8&5f2J
")630-0101 F‘NI8)@86p;T9b;3D.8916-5
{yuuta-t,matsu}@is.aist-nara.ac.jp

K\8&5fGO!"5!3#3X=,<jK!(Support Vector Machines) $rMQ$$$F%a!<%j%s%0%j%9%H$NCx<T<1JL$r9TC?!#$^$?!"%a!<%j%s%0%j%9%H$N%G!<%?$G3X=,7?<1JL4oKhCF!"Web $NJ8=q$NCx<T<1JL$r;n$_k3HG0[Jk%?%$%w$N%I%-%e%a%s%HKBP9$k@-G=$rD4Y?!#3N:]!"=>Mh$+$i;Hol$F$$$?C18lN-gram HHbK!"%G!<%?%^%$%K%s%0<jK!(PrefixSpan) KhCFCj=P5l$?C18l$NO"B3%Q%?!<%s$rAG@-$KMQ$$$k3HGh$j9b$$@-G=$,F@$il?!#;HMQ5l$?%Q%?!<%s$ONY$j9g$o$J$$C18lNsKb%^%C%A9k%Q%?!<%sG"$j!"<B37k2L$h$jCx=R%9%?%$%k$rI=8=$9kNKE,EvJFCD'$N0l$DG"kH9M$($ilk!#

%-!<%o!%I:Cx<T?dDj,EE;R%a!<%k,Web J8>Q,%5%]!<%H!&%Y%/%?!&%^%7%s,O"B3%Q%?!<%s!&%^%$%K%s%0,%W%l%U%#%C%/%9!&%9%Q%s

1 Introduction

Until recently, computers did not play important roles in Japanese authorship identification according to Murakami [16], since computers could not handle Japanese characters and word segmentation in Japanese is not trivial. However, these limitations have been resolved for the most part in the recent past.

Traditionally, the objects of authorship identification have for the most part been literary texts. However, as the amount of available computer-readable texts continues to increase, demand is growing for techniques which are more robust and applicable to documents in the wider domains. One of these demands is in forensic linguistics. According to Chaski [4], there are many different types of crime and civil action involving documents whose authorship has to be authenticated.

To achieve robust authorship identification, we introduce sequential word patterns as a feature for classification purposes using support vector machines.

In section 3, we discuss sequential word patterns as a style marker of authorship.

There are many studies on the statistical analysis of the style of a particular author, called stylometry, using a variety of quantitative criteria. Word or sentence length, vocabulary richness, word ratios, and part of speech ratios have been used in conventional studies [10]. However, it is still an open question which style markers are more appropriate.

Yoshida et al. [5] compared the effectiveness of various features in authorship identification for Japanese literary texts. Their result suggests that lexical N-grams (trigram in their case) are the most effective features for authorship identification, and that character trigrams, distribution of characters before commas, and *Hiragana* usage in each line are also effective. Still, the length of N-grams is not flexible, and the sequence of words are always contiguous in the conventional N-gram model.

In the present research, we employ a sequential pattern mining technique to overcome these limitations (section 2). Therefore, not only rigid segments of lexical items, but also flexible (non-contiguous) lexical items can be considered as features for authorship identification.

In section 4, we introduce a state of the art machine learning algorithm to the task of author identification. Although there is no clear agreement on the style markers, conventional techniques rely on a few carefully selected features in order to avoid the *curse of dimensionality* [3]. Recently, the machine learning community has paid much attention to large margin classifiers, which have theoretically good generalization capability independent of the dimensionality of the input space. We applied one such large margin classifier, namely support vector machines, for authorship identification in order to consider both conventional word N-gram features and the sequential word pattern features we propose, the dimensionality of which is considerably high.

In section 6, we show the result of authorship identification experiments conducted on E-mails and Web documents.

2 Sequential Pattern Mining

2.1 Problem Statement

Agrawal and Srikant [2] introduced the sequential pattern mining problem which is formulated as following. Let $I = \{i_1, i_2 \cdots, i_m\}$ be a set of literals, called *items*. An *element* is a non-empty set of items. A *sequence* s is an ordered list of elements denoted by $< s_1, s_2, \cdots, s_l >$ where s_j is an element. The number of instances of *items* in a sequence is called the *length* of the sequence. When the item set of an element a is the subset of or equal to the item set of an element b, we denote $a \subseteq b$. A sequence $\alpha =< a_1, a_2, \cdots, a_n >$ is a subsequence of another sequence $\beta =< b_1, b_2, \cdots, b_m >$ if there exist integers $1 \leq j_1 < j_2 \cdots < j_n \leq m$ such that $a_1 \subseteq bj_1, a_2 \subseteq bj_2, \cdots, a_n \subseteq bj_n$.

A *sequence database* S is a set of tuples $< sid, s >$ where sid is a *sequence_id* and s is a sequence. A tuple $< sid, s >$ is said to contain a sequence α, if α is a subsequence of s (i.e., $\alpha \sqsubseteq s$). The support of α is the number of tuples in the database containing α and is denoted $support_S(\alpha)$. Given a positive integer ξ as the *support threshold*, a sequence α is called a (frequent) *sequential pattern* in a sequence database S if the sequence is contained by at least ξ tuples in the database; in other words, $support_S(\alpha) \geq \xi$.

Given a sequence database and a user-specified *minimum support* threshold, the problem of *sequential pattern mining* is to find all sequences whose support is greater than or equal to the *minimum support*.

In section 3, we discuss *sequential word patterns* based on this sequential pattern mining problem where *item* and *sequence* correspond to *word* and *sentence*, respectively. Note that, we can assume that each *element* has a single item (word) in our application, we denote *item* as an element of a sequence.

2.2 The PrefixSpan Algorithm

Most conventional methods for mining sequential patterns are based on *Apriori* [1] property, which states that any super pattern of a non-frequent pattern cannot be frequent. These *Apriori*-like methods adopt the *candidate-generation-and-test* approach. In this approach, each subsequent pass generates candidate sequences, counts their supports by scanning a sequential database, and prunes the candidates whose support

is less than the minimum support threshold. The next pass generate new candidates which have one more item than the set found in the previous pass. The algorithm terminates when no new sequential pattern is found in a pass, or no candidate sequence can be generated.

The performance of the *Apriori*-like sequential pattern mining method is degraded by the generation of the huge set of candidates and the multiple scans of whole database necessary when mining a database containing long and/or voluminous sequential patterns.

Pei et al. [6] introduced a divide-and-conquer approach for the sequential pattern mining problem, called *PrefixSpan* (Prefix-projected Sequential pattern mining).

PrefixSpan narrows the search space based on the prefix of sequential patterns in a recursive manner. It uses each frequent item to partition the sequential database into a set of smaller databases sharing the item as the prefix of patterns to be found. Then, it recursively searches for frequent subsequence patterns in each smaller database. We define these sub-databases, called *projected database*, to explain the PrefixSpan algorithm.

Definition: Let α be a sequential pattern in a sequence database S. The *α-projected database*, denoted as $S|_\alpha$, is the collection of postfixes of sequences in S which have prefix α.

The PrefixSpan algorithm is based on the following lemma on projected databases.

Lemma: Let α and β be sequential patterns in a sequence database S, and b is an item such that α is a prefix of β; in other words, $\beta = \alpha b$.

1. $S|_\beta = (S|_\alpha)|_b$
2. for any sequence β having prefix α, $support_S(\beta) = support_{S|_\alpha}(b)$
3. The size of an α-projected database cannot exceed that of S.

Then, the PrefixSpan algorithm is described as follows:

Algorithm:

Input A sequence database S and the minimum support threshold ξ

Output The complete set of sequential pattens with frequency no less than ξ

Method Call *PrefixSpan*(<>, S)

Subroutine *PrefixSpan*(α, $S|_\alpha$)

- Parameters:
  α: a sequential pattern, $S|_\alpha$: α-projected database.
- Method:
  1. Find a set of items B whose element b is such that $support_{S|_\alpha}(< b >) \geq \xi$
  2. For each item $b \in B$,
     (a) Append b to α to form an extended sequential pattern αb and output it
     (b) Construct the αb-projected database $(S|_\alpha)|_b$ for each αb and call *PrefixSpan*(αb, $(S|_\alpha)|_b$)

3 Sequential Word Patterns

We applied PrefixSpan to extract sequential word patterns from each sentence and used them as author's style markers in documents. The *sequential word patterns* are sequential patterns where *item* and *sequence* correspond to *word* and *sentence*, respectively.

The sequential word patterns are denoted as $< w_1 * w_2 * \cdots * w_l >$ where w_i is a word, l is the length of sequential pattern, and $*$ is any sequence of words including the empty sequence.

These sequential word patterns were introduced for authorship identification in the present research based on the following assumption. Because people usually generate words from the begging to the end of a sentence, how one orders words in a sentence can be an indicator of author's writing style. As word order in Japanese is relatively free, rigid word segments and not-contiguous word sequences may be a particularly important indicator of the writing style of authors. Experimental evidence will be provided in section 6.1.

Takeda et al. used sequences of adjuncts (auxiliary verbs and postpositions) to uncover characteristics of *Waka*, a traditional style of Japanese poetry. Such sequences of adjuncts are a special case of sequential word patterns in which the items of the sequences are limited to adjuncts. They reported successful results in finding patterns from five anthologies. Previous usage of restricted versions of sequential word patterns support this assumption.

Note that, because we identify the author of a document (not a sentence) in this research, it is more intuitive that the support of sequential word patterns is calculated by counting the number of documents including them. Thus, the sequence database and the support count are redefined as follows.

Definition (Sequence Database and Support Count based on Documents): A sequence database S' is a set of tuples $< docid, sid, s >$ where $docid$ is document_id, sid is a sentence_id and s is a sequence of lexical items which represents a sentence. Let α be a sequential pattern in a sequence database S', and β be a sequence having prefix α. The support count of β on α-projected database $S|_\alpha$, denoted as $support_{S|_\alpha}(\beta)$, is the number of documents including sequences γ in $S|_\alpha$ such that β is composed of the prefix α and the postfix γ.

Although N-grams may partially cover the features derived from the above, there are two advantages in employing a sequential pattern mining technique over a conventional N-gram model.

One advantage is that the mining method can handle flexible (non-contiguous) word sequences. N-grams are consecutive word sequences and fail to account for non-contiguous patterns. The sequential pattern mining

overcomes these limitations, since it can cope with any number of intervening words in a sequence.

Another advantage of the sequential mining technique is that there are fewer user-specified parameters. Usually, infrequent N-grams are omitted with an arbitrary threshold (i.e. a minimum support in the data mining terminology). At the same time, the length of N-grams is arbitrarily decided by users, although we never know how long the actual length should be. Thus, we need to specify two parameters, the threshold and the length. On the other hand, the sequential pattern mining does not limit the length of sequential patterns as long as they are frequent. The sequential pattern mining requires only a single parameter, the *minimum support threshold*.

The sequential word patterns defined here are extracted by the PrefixSpan algorithm and their frequencies of them are employed as the document features of an author.

4 Support Vector Machines

Because authorship identification can be considered to be a categorization problem, we employ support vector machines, which achieve state of the art accuracy in the categorization of document topics.

Support vector machines (SVM) are one of the large margin classifiers attracting attention in the machine learning community. According to the principle of structural risk minimization [13], minimizing the model complexity (structural risk) and training error (empirical risk) leads good generalization (i.e. good performance for test data never seen in the training data).

This minimization process is embedded in SVMs by constructing a hyperplane as the decision surface such that the margin between the positive and the negative examples is maximized.

Consider the training samples $\boldsymbol{S} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_l}, y_l)\}$, where $\boldsymbol{x_i}$ is the vector representing an input example, and y_i is its labels. The equation of a decision surface in the form of a hyperplane that does the separation is $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ where $\boldsymbol{w}$ is an adjustable weight vector, and b is a bias. The margin of the optimal hyperplane is derived from $2 \;/\; \|\boldsymbol{w_o}\|$ where $\boldsymbol{w_o}$ denote the optimum values of the weight vector. Thus, maximizing the margin is equivalent to minimizing the Euclidean norm of the weight vector $\boldsymbol{w}$.

This maximal margin strategy allows support vector machines to have a generalization capability independent of the number of model parameters. Theoretically, a SVM classifier can perform well on unknown data even with a high dimensional input space. This property of SVMs is suitable for authorship identification. Since there is no consensus on the effective style markers, all possible features having potential usefulness can be used so that the input space tends to be high dimensional.

Joachims [8] used the SVMs for text categorization tasks and achieved good performance with nearly 10 thousand features.

Diederich et al. [7] applied SVMs to identify the authors of newspaper articles. They compared usage of all words to bigrams of part-of-speech tags and functional words, and conclude that the full words perform better than the bigrams.

De Vel et al. applied SVMs for authorship identification of E-mails [11]. They generated a controlled set of E-mails for each author and topic, and showed the good identification performance independent of the topic of the E-mails, even though they used only conventional style marker features (e.g. the vocabulary richness).

Author	Addresses	Messages
A	3	1675
B	3	392
C	1	335

Table 1: Number of E-mail Address used and Number of Messages posted by Target Authors

5 Experimental Methodology

5.1 E-mail and Web corpora

To show the effectiveness of our approach, we conducted two experiments. One is authorship identification of E-mail messages and the other is the authorship prediction of Web documents based on the E-mail message information.

In the first experiment, we employed 4961 Japanese messages from a mailing list of a computer programming language development community. The characteristics of this E-mail message corpus are as follows: The length of the documents tends to be short (the average length of the messages is 112 words), and a single author usually writes the message body excluding citations. The messages are sent from 111 distinct Internet mail addresses. The messages were sent during a time span of 562 days.

Three frequent senders were chosen for the identification experiment. Table 1 shows the number of E-mail addresses which the target authors used (*Addresses*), and the number of their posts to the mailing list (*Messages*).

This corpus is used to determine the authorship identification performance when employing the same type of documents both for training and testing.

In the second experiment, to determine the capability of identifying authorship for heterogeneous documents,

Author	Documents	Positive	ML
A	547	74	0
B	501	439	92
C	501	407	200

Table 2: Web corpora: *Documents* stands for the number of documents in the corpus, *Positive* stands for the number of documents written by the target author in the *Author* column, *ML* stands for the number of documents which are the messages posted on the same mailing list of the E-mail corpus.

we use Web documents, which are retrieved by a Web search engine, Google[1]. An SVM classifier is trained on the mailing list messages mentioned above, and applied the classifier in order to identify the author of Web documents.

We prepared a different Web corpus for each author in the following way. The authors' Japanese full names which were manually extracted from the E-mail message corpus are submitted as a query to the Web search engine. For each target author, we downloaded about 500 HTML documents which the search engine retrieved as the most relevant documents for the name. These documents were manually tagged as to whether they were written by each target author or not.

The characteristics of these Web corpora are stated as follows: The length of the documents are longer than that of E-mail messages (the average length of the documents is 663 words), and the documents written by multiple author are common so they tend to be noisy data.

Table 2 shows the number of Web documents which are labeled as positive examples; in other words, the documents written by a target author.

The positive documents include software manuals, messages on Web Bulletin Board Systems, the home pages of the target authors, Web diaries, mailing list messages publicly distributed on the Web, and so on. The *ML* column in Table 2 shows the number of documents which were actually posted on the same mailing list as the E-mail corpus. Since none of these messages had been included in the E-mail corpus, and because they still included commercial messages and/or navigation texts which are observed in Web documents, these documents have different properties from the E-mail corpus. Thus these messages were included in the Web corpora.

As preprocessing of both corpora, citation (marked with $>$, $|$, etc.) removal and sentence segmentation were performed using pre-specified rules. The morphological analyzer, ChaSen version 2.2.8 [15], was used for word segmentation and part of speech tagging.

[1] <http://www.google.com/>

5.2 Performance Measures

In the experiment with the E-mail corpus, we employed a cross-testing procedure. The original message set was divided into 5 subsets of nearly equal size. Then, five different SVMs were trained on 4 of the subsets and the remaining one subset was used for testing. That is to say, about 4000 messages ware used for each training and about 1000 messages for each testing.

To evaluate identification performance, we calculate accuracy, precision, recall, and F-measure value, which are metrics commonly used to evaluate information retrieval and text categorization performance [9]. Accuracy is the percentage of correctly classified examples so that it describes the classification performance of both the positive and the negative examples. Precision is the percentage of the positively classified examples which are actually positive. Recall is the fraction of positive examples which has been classified as positive. F-measure combines precision (P) and recall (R) scores into a single value using the formula:

$$F = \frac{2PR}{(P+R)}$$

The F-measure describes the extraction capability of positive examples.

Since the SVMs are binary classifiers, we averaged the identification performance of all target authors (i.e. all classifiers) using macro-averaging [14]. Macro-averaging gives an equal weight to the identification performance of every target author.

5.3 Features and The SVM Classifiers

The features used in the experiments were the frequencies of word N-grams and the sequential word patterns described in section 3.

For word N-grams, we employed the union of unigrams, bigrams, and trigrams. Since the number of distinct bigrams and trigrams is large, we employed only the bigrams and trigrams which appeared in more than 2 documents. The number of distinct elements of this union is 58064, so that the input space of the classifier has very high dimension. The value of each feature is term frequency dampened by log function.

For sequential word patterns, we applied the PrefixSpan algorithm described in section 2.2 to extract them from the E-mail corpus. A *longest sequential pattern* is the set of frequent sequential patterns which have no super-pattern among the frequent sequential patterns. To avoid redundancy, we employed only the sequential patterns which the PrefixSpan algorithm outputs if a projected-database $(S|_\alpha)|_b$ cannot construct or does not include any item c such that $support_{(S|_\alpha)|_b}(< c >) \geq \xi$ in step 2 (see. section 2.2), though the PrefixSpan algorithm outputs all the sub patterns of the longest

	Author A		Author B		Author C	
	20%	10%	20%	10%	20%	10%
Runtime(min)	54	164	4	13	2	5
Patterns	351	3070	317	2030	192	949

Table 3: The Result of Sequential Pattern Mining

Pattern	Frequency
$0 * $s * G9 * !% ha * n * desu * !%	0.81(213/261) (TP * N * ESE)
$N * $O * $O * G9 * !% no * ha * ha * desu * !%	0.80(163/202) (of * TP * TP * ESE)
$G * $O * $J$$ * G7$g * $& * $+ * !# de * ha * nai * deshou * u * ka * !#	0.66(43/65) (ESE, asking a agreement)

Table 4: Examples of Sequential Word Patterns; *ESE* stands for end-of-sentence expressions, *N* stands for nominalizers in Japanese, *TP* stands for topic particles in Japanese

sequential pattern. For each target author, two sets of sequential word patterns were employed based on different minimum support counts: patterns appearing in 20% ore more the messages written by the target author written (i.e. the minimum support is set to 20%), and patterns appearing in 10% or more.

To implement the PrefixSpan algorithm, we employed the pseudo-projection procedure [6] which uses pointers referring to the sequences in the database, instead of constructing physical projections by collecting all the postfixes. The PrefixSpan algorithm is implemented in Ruby[2], executed on Linux (Pentium III 900 MHz). We concatenated those sequential word patterns with the union of N-grams as previously described.

As mentioned in section 4, the SVM classifiers were employed in the experiments. A SVM implementation, TinySVM [12], was used with linear kernel function and the balance parameter value, C, was set to 1.0.

6 Results and Discussion

6.1 Sequential Pattern Mining

Table 3 shows the runtime and the number of patterns found for each target author. The number of found patterns with minimum support 10% is many times over that with minimum support 20%.

Table 4 shows the examples of sequential word patterns which were extracted from the Japanese E-mail corpus. Frequency stands for the ratio of documents written by a particular author including the sequential word pattern over all documents including the pattern. They provide evidence supporting the assumption described in section 3: 80% and 66% of the appearances of the example sequential word patterns occurred in the documents written by a target author.

6.2 Authorship Identification for E-mails

Table 5 shows the authorship identification results of E-mail messages. The column "123" stands for the union of word unigrams, bigrams, and trigrams. The column "123s20" stands for the features of the "123" column plus sequential word patterns with a minimum support of 20% and the column "123s10" stands for the features of the "123" column plus sequential word patterns with a minimum support of 10%.

SVMs achieved high identification performance despite the large number of features. Although the effect of the sequential word pattern features may not seem obvious, the result shows better F-measure value; in other words, the classifier with the sequential word pattern features could find more messages of the target authors than the one without sequential word pattern features.

6.3 Authorship Identification for Heterogeneous Documents

Table 6 shows the authorship identification result applied to the Web documents. The label of each column is the same as the ones for table 5.

Contrary to the previous result, the effect of the sequential word pattern features is evident. In all measures, the application of sequential word patterns outperforms that of N-grams alone. In particular, author *A*'s identification performance shows significant improvement. None of the Web documents for author *A* is of the same type as the messages in original E-mail corpus (see Table 2) so that all the documents have different properties from the training data. This makes the identification of *A* more difficult than the others. Thus, we conclude that sequential word patterns for authorship identification are especially effective in cases in which the type of test data is different from the training data.

7 Conclusion

In this paper, we proposed the use of support vector machines which have good generalization capability, and new style markers of authorship, namely frequent sequential word patterns, which match both rigid and flexible word sequences. To extract these patterns, we applied a sequential pattern mining technique, PrefixSpan.

Firstly, we viewed the authorship identification tasks on E-mail corpus, and achieved good performance using SVMs with the input space of high dimensionality.

[2] <http://www.ruby-lang.org/>

	Accuracy			Precision			Recall			F-measure		
Author	123	123s20	123s10	123	123s20	123s10	123	123s20	123s10	123	123s20	123s10
A	**98.50**	98.46	98.44	96.57	**98.75**	96.75	**98.53**	98.20	98.14	**97.54**	97.47	97.44
B	99.44	99.36	**99.45**	97.69	97.34	**97.75**	94.32	93.64	**94.75**	95.98	95.46	**96.23**
C	**98.03**	97.87	97.99	**83.52**	82.08	83.39	87.20	87.13	**87.61**	85.32	84.53	**85.45**
Average	**98.66**	98.56	98.63	92.59	92.06	**92.63**	93.35	92.99	**93.50**	92.95	92.49	**93.04**

Table 5: The Identification Result of E-mails

	Accuracy			Precision			Recall			F-measure		
author	123	123s20	123s10	123	123s20	123s10	123	123s20	123s10	123	123s20	123s10
A	78.4	84.8	**87.2**	26.5	44.3	**53.3**	33.7	**47.2**	43.2	29.0	45.4	**47.4**
B	89.4	89.6	**91.0**	97.3	**97.7**	97.5	90.4	90.2	**92.0**	93.3	93.3	**94.4**
C	53.0	66.4	**67.0**	**92.5**	91.9	92.3	45.9	64.3	**64.8**	60.4	75.1	**75.4**
Average	73.6	80.3	**81.7**	72.1	78.0	**81.0**	56.7	**67.2**	66.7	60.9	71.3	**72.4**

Table 6: The Identification Result of Web documents

Next, to examine the identification capability for heterogeneous documents, the SVM classifiers which trained on E-mail corpus were applied to the author identification of Web documents. Experimental results showed that the union of conventional N-grams, and sequential word patterns achieved better performance than N-grams alone with heterogeneous documents.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pp. 487–499. Morgan Kaufmann, 12–15 1994.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Philip S. Yu and Arbee L. P. Chen, editors, *Proc. 11th Int. Conf. Data Engineering, ICDE*, pp. 3–14. IEEE Press, 6–10 1995.

[3] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[4] Carole E. Chaski. Linguistic authentication and reliability. In *National Conference on Science and the Law*, San Diego, California, April 1999. National Institute of Justice.

[5] Atsuhiro Yoshida et al. Effective features of authorship identification. In *IPSJ SIG Notes*, No. 2001-FI-64, 2001-NL-145, pp. 83–90, 2001.

[6] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. 2001 Int. Conf. on Data Engineering (ICDE'01)*, pp. 215–224, Heidelberg, Germany, April 2001.

[7] Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. Authorship attribution with support vector machines. poster presented at The Learning Workshop, April 2000.

[8] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pp. 137–142, 1998.

[9] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[10] Tony McEnery and Michael Oakes. *Authorship Identification and Computational Stylometry in Handbook of Natural Language Processing*, chapter 23, pp. 545–562. Marcel Dekker Inc, 2000.

[11] O. De Vel, A. Anderson, M. Corney, and G. Mohay. Mining Email Content for Author Identification Forensics. SIGMOD: Special Section on Data Mining for Intrusion Detection and Threat Analysis, December 2001.

[12] Taku Kudo. *TinySVM: Support Vector Machines*. <http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM>, 2001.

[13] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. New York, 1995.

[14] Yiming Yang. An evaluation of statistical approaches to text categorization. Technical report, Carnegie Mellon University, 1997.

[15] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka and Masayuki Asahara. *Morphological Analysis System ChaSen Manual*. Nara

Institute of Science and Technology, version 2.2.8, 2001.

[16] B<>e@,>!. 2r@b !J8>O$N7WNLJ,@O!]$=$NNr;K$H 8=>u!]. 7WB,$H@)8f, Vol. 39, No. 3, pp. 216–222, March 2000.