Temporal Context for Authorship Attribution
A Study of Danish Secondary Schools

Niels Dalum Hansen[1] ✉, Christina Lioma[1], Birger Larsen[2], and Stephen Alstrup[1]

[1] Department of Computer Science, University of Copenhagen, Denmark
[2] Department of Communication, Aalborg University, Denmark

Abstract. We study temporal aspects of *authorship attribution* - a task which aims to distinguish automatically between texts written by different authors by measuring textual features. This task is important in a number of areas, including plagiarism detection in secondary education, which we study in this work. As the academic abilities of students evolve during their studies, so does their writing style. These changes in writing style form a type of temporal context, which we study for the authorship attribution process by focussing on the students' more recent writing samples. Experiments with real world data from Danish secondary school students show 84% prediction accuracy when using all available material and 71.9% prediction accuracy when using only the five most recent writing samples from each student.
This type of authorship attribution with only few recent writing samples is significantly faster than conventional approaches using the complete writings of all authors. As such, it can be integrated into working interactive plagiarism detection systems for secondary education, which assist teachers by flagging automatically incoming student work that deviates significantly from the student's previous work, even during scenarios requiring fast response and heavy data processing, like the period of national examinations.

Keywords: Authorship attribution, secondary education, automatic classification

1 Introduction

Given various documents written by different authors, *authorship attribution* is an automatic classification task that aims to identify which documents are written by the same author. Attributing authors to text is an important task, with applications to email forensics for cybercrime [7], literature [22], or education [1], for instance. An example of the latter, which we study here, is the application of authorship attribution for plagiarism detection in secondary education. Increasingly more secondary education institutes (roughly corresponding to ages fourteen to eighteen in most countries, e.g., high schools) use digital learning software to manage the submission, grading, feedback and assessment of students. Tasks and processes traditionally carried out manually by teachers are

increasingly now automated in order to assist teachers. One of those tasks is the flagging of student work on the grounds of plagiarism, usually done by the student copying another student's work or material found in books or online. Authorship attribution methods are commonly used in digital learning management systems to detect student submissions whose writing style seems to depart from the student's previous work. These systems build and continuously update an *authorial fingerprint* for each student, based on all his incoming writings; when a newly submitted document deviates notably from the author's fingerprint, the system alerts the teacher.

There are two main challenges associated with the task of authorship attribution in the domain of plagiarism detection for secondary education. First, the writing style of students often evolves rapidly during their formative years at school, especially when students with poorer writing skills make an effort to improve their writing. This means that what a student writes at the age of fourteen may not be a good indicator of his writing style by the age of seventeen. To deal with this, authorship attribution systems require special adjustment to the course-specific intended learning objectives and expected progress of the students. Practically, most digital learning systems lack this, and manual inspection by the teachers is required. The second challenge is that very often, the amount of text stored in each student's profile is limited, with the resulting data scarcity having a notable impact upon classification accuracy. This data scarcity is due to the fact that, even if a digital learning system logs data on thousands of students over many years, the writings of each student are unique, and can only consist of the total number of course-specific assignments, projects and exams written by that student. For instance, in our study, even though we have access to the writings of >100,000 students (see Section 3.1), each student produces per year on average 4.2 documents of approximately 6700 characters each, per course. For a 3-year education, this means that the data available for classification is on average only 12 documents (approximately 80,400 characters) per student.

The above challenges motivate this work. Even though it is generally agreed that the more data available for classification, the higher the accuracy (or the lower the error of the estimate) [14], we set out to examine whether adequate classification accuracy can be expected with fewer more recent data, in the domain of authorship attribution for secondary education students. To our knowledge, there is not much empirical material on this aspect of authorship attribution for this domain.

In the rest of this paper, related work is overviewed in Section 2; our study design, including data, is presented in Section 3; the findings are discussed in Section 4; conclusions and future research are outlined in Section 5.

2 Related Work

Historical and temporal aspects have long been recognised as an important contextual feature in information retrieval (IR). While they are mostly ignored in test collection-based *ad hoc* IR research, studies in interactive IR often in-

clude temporal aspects. For instance, Kumpulainen and Järvelin [12] carried out longitudinal studies of molecular medicine researchers in work tasks involving diagnosis, prevention and treatment of various diseases. They found time to be a central contextual feature for recording and understanding the complexity of the between-systems interaction, and also an important factor in proposing solutions for optimising information access. More broadly, the concept of time in information interaction has consistently been mentioned as an element that needs to be considered both theoretically and practically, see for instance Ingwersen and Järvelin [6] for a discussion of time in relation to most major interactive IR components. Temporal context is further emphasised in Ingwersen (2008), where a separate historical context dimension is added to a stratified model of context types forming part of an integrated research framework for interactive IR [5]. Historical context in this case means the history of all participating actors' experiences. Ingwersen notes that the historic context operates across this stratification of contexts and that all interactive IR processes and activities are influenced by this form of temporal context (page 7 in [5]).

In this study we focus on the temporal context of authors, i.e. of the student writing, which we believe affects their writing behaviour. We examine this context for classification, not retrieval, and specifically for authorship attribution. Several authorship attribution methods have been presented. A somewhat outdated review of authorship attribution methods can be found in Juola [8]; see Koppel et al. [9], Savoy [15, 16], and Stamatatos [21] for more recent overviews (in the last five years). Generally, authorship attribution methods are typically assessed on comparative predefined (TREC[3]-style) datasets. Among the best performing methods of such evaluations are support vector machines (SVMs), which are reported to reach a classification accuracy approximating 91% on specific tasks and datasets [8]. A performance-boosting technique sometimes used with SVMs is the combination of two feature sets, a *primary* consisting of so-called 'unstable' words (i.e. words that can easily be replaced by a synonym), and a *secondary*, consisting of the k most frequent words in the training data. If the primary feature set does not result in accurate predictions, the secondary is used. An interesting comparison of some of the main features used for authorship attribution can be found in Savoy [17]. Note that in Savoy [17], document frequency appears most discriminative for authorship attribution in the domain of sports journalism. Overall, function words (i.e. stopwords) are one of the more robust and more discriminative features [8].

Alternative competitive approaches to SVMs for authorship attribution are (i) Latent Dirichlet Allocation (LDA), e.g. the recent LDA extension that includes author-aware topic models [18]; and (ii) the data compressor-based approach of de Oliveira et al. combined with the Normalized Compression Distance (NCD) document similarity metric [2], which is reported to outperform SVMs on certain datasets.

Most authorship attribution research focuses on few authors with a large amount of written material [19]. This scenario, apart from not being always

[3] Text Retrieval Evaluation Conference, see http://trec.nist.gov

realistic as for instance in our case (see the discussion in Section 1), also risks overestimating the importance of the features extracted from the training data that are found to be discriminative for these small sets of authors [13]. In our work this situation is almost reversed, as the task requires us to look at many authors with small amounts of written material available. This scenario of a large number of authors with often very little text for each author has been referred to as *authorship attribution in the wild* [9] or *authorship verification* [13], and is known to be particularly challenging.

Another difference of our work to previous research is the focus on secondary education students, which might make for less predictable sources of writing styles than established authors. The reason for this assumption is two-fold: on one hand, students change both with regards to personality and writing style during their formative years in school; on the other hand, secondary school students are less trained at writing than adult or professional writers. These two reasons make authorship attribution in this domain rather challenging, and motivate us to look into the effect of using more recent writing samples from the students for classification.

3 Study Design

Our study aims to examine the practical feasibility of using limited and recent writing samples from students for authorship attribution in the domain of plagiarism detection for secondary schools. We next describe our data and methodology.

3.1 Student data

Our data is supplied by MaCom[4], a company specialising in providing ERP (Enterprise Resource Planning) systems to secondary schools in Denmark. Specifically the data comes from Lektio, a system for managing digital student submissions, which is used by 9/10 secondary schools in Denmark. We were given access to a database of >100.000 students, which consisted of anonymised student IDs and associated data submitted by each student (e.g. the text of their assignment, feedback, project, exam) and related meta-data (e.g. dates, grades, courses followed). From this database, we extracted randomly the profiles of 30 students, using these filters: (i) duplicate removal, and (ii) student performance and study duration. We describe these filters next.

Filter 1: Removal of duplicate documents. Initial discussions with MaCom revealed their experience that student profiles may sometimes contain the same document twice, with no or minor modifications. Such cases of (near-)duplicate documents risk compromising the accuracy of authorship attribution methods by over-emphasising the discriminative power of certain features artificially, for

[4] http://www.macom.dk

instance. Therefore, we decided to detect and remove identical documents from the dataset. To this aim, we calculated the linguistic cross-entropy between all documents, and then empirically found the cross-entropy score which seemed to separate identical documents from similar documents. We explain this next.

Entropy can be used as a measure of information content or randomness in a stream of symbols [20]. Given a communication channel M, which emits messages $m_1, m_2, ..., m_n$, each with probability $p_1, p_2, ..., p_n$, the most efficient encoding of stream M, i.e. the shortest encoding where all messages are distinguishable, is obtained when the length of the binary encoding of m_i is equal to $-log_2(p_i)$. Based on this, the entropy H of M is the number of bits needed to make a unique encoding of each message when the shortest encoding is used:

$$H(M) = \sum_i (-log_2(p_i)) \times p_i \tag{1}$$

In practice the distribution of the messages might not be known and is therefore estimated from samples. From the above definition of entropy, we can see that the entropy depends on both the encoding length $log_2(p_i)$ and the frequency of the message p_i. This gives rise to the definition of cross-entropy:

$$H(M) = \sum_i (-log_2(q_i)) \times p_i \tag{2}$$

where q is an estimated distribution and p is a stochastic model of the communication channel. Following [3], computing the cross-entropy of two documents is based on counting the match length of a string representing a series of messages, within a database (also a string) representing a sample of messages from the source M. More formally, the match length L of a string S with length $|S|$, in the database D with length $|D|$ is defined as:

$$L = max\{l : 0 \leq l \leq |S|, S_0^{l-1} = D_j^{j+l-1}\} \tag{3}$$

for some $j : l - 1 \leq j + l - 1 \leq |D|$. The match length is found for all indices of S. Letting $\bar{L}$ denote the average match length for all indices of S, it is possible to estimate the cross-entropy ($\hat{H}$) as:

$$\hat{H} = \frac{log_2(|D|)}{\bar{L}} \tag{4}$$

Formally, some assumptions are violated by computing cross-entropy on our data: first, the events in the stochastic process, e.g. the communication channel M or in our case the documents written by secondary school students, are not uniform i.i.d. (independent and identically distributed); and second, because student writing evolves with time, the stochastic process is not a stationary, ergodic source with finite memory. Despite these violations of Equation 4, we assume that cross-entropy can still give meaningful results for duplicate detection.

After computing cross-entropy, we ranked all documents in ascending order according to their cross-entropy score. The first 70 results were manually inspected and in all of them only two seemed to be plagiarised. The remaining could be categorised into the following cases:

1. Documents belonging to *group work*, where each group member submits a copy of the same document.
2. *Duplicate upload*, i.e. the same document is uploaded by the same student within a few seconds interval.
3. Documents with *teacher comments*, as part of the teacher's feedback to the student.
4. *Resubmission* of almost identical documents by the same student with a interval of a few weeks, for example as part of a student portfolio.

We manually inspected this data and defined as duplicate documents to be removed those documents that (i) belonged to a pair of documents written by the same author, and (ii) were the oldest in a document pair with a cross-entropy score lower than 1. We applied this decision to remove such assumed duplicates automatically from the whole dataset.

Filter 2: Student performance and study duration. To ensure no bias in favour or against high/low-performing students, we randomly selected fifteen low-grade students and fifteen high-grade students. Specifically, the low-grade students have a grade point average of 3.72 in the Danish grading scale (see Table 1 for equivalences to the 100% grading scale) and the high-grade students have a grade point average of 11.02. The individual grade point averages of the 30 students in our dataset are displayed in Table 3.

The writings of each student cover all three last years of their secondary school education (corresponding to ages fifteen to eighteen), and specifically from August 2009 to May 2012, evenly distributed. This selection (and duplicate removal) resulted in a dataset of 405 documents in total for 30 students, as shown in Table 2. We assume that this dataset reflects the general distribution of writing that high-grade and low-grade students produce for those years of their secondary education.

Table 1. The Danish 7-point grading scale and its approximate correspondence to the 100% grading scale. The Danish grades are fixed to these 7 points, i.e. no decimals are allowed. The decimals reported in this paper are due to averaging the grades of several students.

7-point scale	100% scale	grade explanation
12	100-90	excellent
10	89-80	very good
7	79-70	good
4	69-60	fair
02	59-50	adequate
0	49-40	inadequate
-3	39-0	unacceptable

Table 2. Number of documents (*docs*) and their average character length in our dataset after duplicate removal.

high-grade student docs	219 (av. length: 6817 char.)
low-grade student docs	186 (av. length: 4656 char.)
total docs	405 (av. length: 5825 char.)

3.2 Classification with Support Vector Machines (SVMs)

We use machine learning, and specifically SVMs to attribute authors to the documents in our dataset. SVMs are binary classifiers; classification is done by treating the training data instances as points in a n dimensional space and then finding a hyperplane separating the two classes. Since our classification task is multi-class, i.e. there are more than two candidate authors, our classification problem has to be reformulated as a binary classification task. This is typically done using an *one-against-all* method, where a separate classification problem is formulated for each author: for each student-author s, a dataset is generated where the documents by s are labelled 1 and all other documents are labelled -1. For each of the authors, a separate test set, training set and a separate model is then generated.

We use two different SVM set-ups, one with a linear kernel and one with a Gaussian kernel. Selecting the most appropriate kernel depends heavily on the problem at hand. A refined and informed selection of kernels for our task is outside the focus of this study. We use the Gaussian kernel because it is a common first choice for SVM classification [4], as it makes it possible to separate non-linearly separable data, while requiring only two hyper-parameters (parameters set by the user). It also has fewer numerical difficulties than other kernel functions, for example kernel values are always between 0 and 1 [4]. When the number of features is very large, the more simple linear kernel often performs better [4], which is why we also use it.

Next we present the features used by the SVMs and our training - testing approach.

Feature selection. SVMs use a feature vector to represent each data instance. A data instance is in this case a student document. The feature vector consists of l real numbers representing properties of the document. The authorial fingerprint of each student consists of all the feature vectors generated from his documents.

Our representation of each document consists of character n-grams extracted from the documents authored by each student. Character n-grams are a known discriminative feature in plagiarism detection [11]. Specifically, we do not use all the character n-grams extracted per document, but only the most common character n-grams as recorded in language (explained below). Practically this corresponds to looking at how often and in which context the student uses the most frequent words in language (stopwords). The use of stopword statistics in this way has been found to be a strong discriminative feature in authorship attribution [17].

We mine the most frequent character n-grams in the Danish language from korpusDK[5], which is a collection of written Danish text (22,013,995 words in 1,287,300 sentences) representing the language used in Denmark around 2000. KorpusDK is maintained by the Danish Society of Language and Literature[6]. We extract all character n-grams, for n=3 (which is one of the default n values of n-grams used for classification [17]), and rank them by their frequency in korpusDK. We then match the character n-grams extracted from our dataset to those extracted from korpusDK, and only keep for a document those n-grams that are among the top k most frequent n-grams in Danish. The value of k is decided in a small experiment described in Section 4 (under *Vector length*).

Cross-fold validation We use a leave-one-out method with 5-fold cross-validation, which is a well-known experimental method aiming to reduce overfitting and strengthen the reliability of our findings. We use a stratified selection to ensure that each class is equally represented in each of the five folds. The grade point average of each student of the stratified selection of the five folds is shown in Table 3. Each of the five folds in turn is used as a test set, while the remaining four are used as a training set. Since SVMs are binary classifiers it is necessary to do the classification as one-against-all. This means that when doing cross-validation, a separate pair of test and training data are generated for each student. In a test and training dataset for a student s, all documents by s have class +1 and all other documents have class -1.

To assess the effect of the student's temporal context upon classification, we modify the leave-one-out method, so that the training set (i) consists of documents older than the test document, and (ii) incorporates a window that controls the number of documents from each student. For a window of size w, the training set consists of w documents from each student; these documents are sorted chronologically from the most to the least recent. I.e. w=1 corresponds to the most recent document, after the test document.

Table 3. Grade point average of each of the six students included in each of the five folds used in cross-fold validation.

Fold	Grade point average
1	3.2, 3.6, 3.9, 10.8, 11.0, 11.1
2	3.3, 3.6, 4.1, 10.8, 11.0, 11.2
3	3.3, 3.8, 4.1, 10.8, 11.0, 11.2
4	3.3, 3.9, 4.1, 10.8, 11.1, 11.2
5	3.4, 3.9, 4.3, 10.9, 11.1, 11.6

Finally, we scale the training data values between 1 and -1, and then perform grid search for best parameters on the training set using 5-fold cross-validation

[5] http://ordnet.dk/
[6] http://dsl.dk

on the training data (details in Section 3.3). The performance of our authorship attribution predictions is evaluated based on the total accuracy for all five folds:

$$Accuracy(S) = 100 \times \sum_{s \in S} \frac{correct[s] + correct[\hat{s}]}{total[s] + total[\hat{s}]} \quad (5)$$

where S is the set of students, $correct[s]$ is the number of correctly classified documents as belonging to s (true positives), $correct[\hat{s}]$ is the number of documents correctly classified as not belonging to s (true negatives), $total[s]$ is the total number of documents by s, and $total[\hat{s}]$ is the total number of documents made by other students than s.

3.3 Technical implementation details

The cross-entropy tests are made using a custom implementation of the cross-entropy estimation technique presented in Farach et al. [3]. The implementation is made in C# compiled with Mono version 2.10.8.1. For classification we use the LIBSVM library version 3.16 [4] (the library is written in C, so we have written a C# wrapper). The code for creating test and training datasets and executing the experiments has also been written in C#. The selection of parameters for the SVMs has been done using a grid search program supplied by LIBSVM. The program has been configured to search C- values in the interval -5 to 15 in steps of 2 and γ-values have been searched in the interval 3 to -15 in steps of -2. Results, student data, korpusDK n-grams and authorial-fingerprints for the SVM and cross-entropy experiments have been saved in a SQLite[7] database, and analysed using scripts written in Python 2.7.

4 Findings

Before we present the findings of the experiments on the impact of temporal context upon authorship attribution for secondary education students, we report two initial mini experiments aiming to guide our decisions regarding (a) vector length and (b) data pre-processing.

4.1 Initial experiments

Vector length. This initial mini experiment aims to guide our decision of vector length, i.e. how many of the top most common character n-grams we should use to build the vectors. As we rank character n-grams by their frequency in Danish, the top k n-grams correspond to the most common words in language[8]. Using, for instance, the top 500 most common n-grams roughly corresponds to using mainly stopwords. We wish to use a number of n-grams that contains

[7] http://www.sqlite.org/

[8] To be precise, as these are character n-grams, not word n-grams, they correspond to character substrings extracted from the most common words.

both common words in language (because they are discriminative for different writing styles) but also common content-bearing non-stopwords (because they are discriminative with respect to the topic of each document). This mixture of common words and content words can be seen as a combination of primary and secondary feature sets, albeit not sequentially as mentioned in Section 2.

To decide the value of k n-grams we will use, we vary the number of n-grams between 1 - 4000, using both the linear and Gaussian kernel on default settings and without pre-processing our data. Figure 1 shows the resulting average accuracy per student. We see that the accuracy tends to follow a whale curve. Performance seems to peak at around 1000 - 1500 n-grams and tippers off slowly. Overall the linear kernel performs better than the Gaussian. The accuracy values are overall lower than reported later because we use default settings. On the basis of this small experiment, we decide to use n-grams at ranks (i) 1000, and (ii) 500-1500. The former choice emphasises the discriminative power of stopwords, while the latter emphasises the discriminative power of document-specific keywords.

Pre-processing. While certain pre-processing steps, like stemming or case-collapsing, may benefit IR, it is not always clear whether they also benefit classification. Especially for authorship attribution, quite often the capitalisation style or preference for certain morphological variants (e.g. participles over prepositional phrases) can be a matter of personal writing style, and hence a discriminative feature in classification. To our knowledge, no study verifying or refuting this has been published for the Danish language, the morphology of which is relatively simple. To get an approximate idea of whether we should pre-process our data or not, we conduct a small classification experiment on one of the five folds of the data (i.e. on a sample of 6 out of 30 students), chosen randomly, using a linear kernel on default settings and the top k =1000 most common n-grams as features.

Table 4 displays the classification accuracy with and without pre-processing (lower-casing, whitespace removal, stemming). We see that, for that fold, no pre-processing yields the best overall performance, slightly outperforming the combination of stemming and lower-casing. Even though this result is reported for only one fold and on default settings, we still consider it as a useful, approximate indication that the performance of our approach will not be greatly compromised by the lack of pre-processing. For the rest of our study, we use no pre-processing.

4.2 Authorship attribution with recent temporal context

We conduct experiments on the whole dataset (30 students) with both the linear and Gaussian kernel, using five-fold cross-validation as described in Section 3.2. For brevity we report only measurements on the linear kernel - the Gaussian kernel yields comparable findings (with overall lower accuracy). Our feature vector consists of the top (i) 1000 and (ii) 500-1500 character n-grams. Figures 2 &

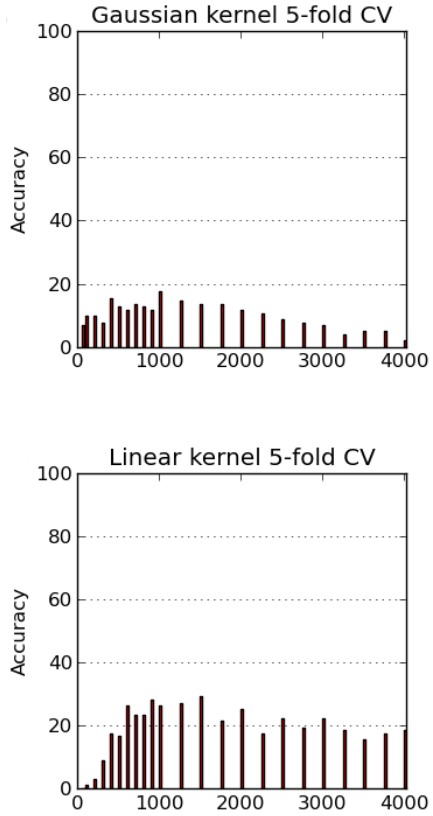

Fig. 1. The x axis shows the number of most common character n-grams used for classification. The y axis shows the respective accuracy. Classification is done with a Gaussian kernel (top) and a linear kernel (bottom).

Table 4. Average classification accuracy for each preprocessing type. The experiments are done on one of the five data folds, using a linear kernel and the top k =1000 most common n-grams.

pre-processing	accuracy
none	40.3
lower-case	38.3
no whitespace	35.0
lower-case + no whitespace	35.0
stemming + low-case	40.2
stemming + low-case + no whitespace	37.3

3 present the classification results when using the top 1000 and 500-1500 most common n-grams, respectively. Each bar represents the accuracy averaged for all documents for all students. The number of documents tested falls for each bar going from 1 to 18, since at bar x, a document by student s is only tested if there are at least x older documents by s to put in the training set. As x increases, the number of documents which fulfil this condition falls.

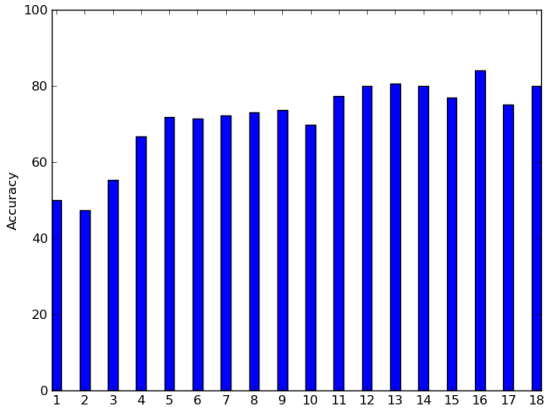

Fig. 2. The x axis shows the average number of documents used for classification per student. The y axis shows the corresponding classification accuracy. The average is taken over all 5 folds. Classification uses the linear kernel and n-grams 0-1000.

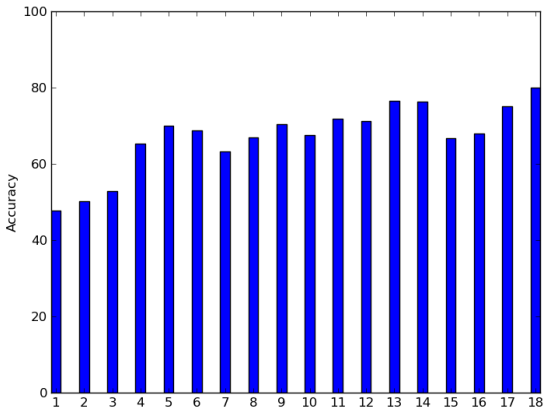

Fig. 3. The x axis shows the average number of documents used for classification per student. The y axis shows the corresponding classification accuracy. The average is taken over all 5 folds. Classification uses the linear kernel and n-grams 500-1500.

Temporal context. We see that using the top 1000 most common n-grams yields better overall accuracy than using the top 500-1500 n-grams. This agrees

with the fact that the use of very common words can be a discriminative feature of personal writing style, as discussed in Section 2. We can report that this fact is also valid for Danish secondary school students.

We also see that using more documents seems to overall improve accuracy. Using, for instance, the 16 newest documents per student yields an accuracy of 84%. Accuracy drops as the number of documents per student is reduced to the most recent, however, interestingly, the drop is not prohibitive: using only the five most recent documents of a student yields an accuracy of 71.9%. Practically this implies that the plagiarism detection system alerts the teacher for cases of potential plagiarism with an error roughly in three out of ten cases. When using all sixteen student documents for classification, not only the five most recent, the respective error rate is two out of ten cases. The change in error rate is not detrimental. Futhermore, the use of fewer recent documents yields notable computational gains, which are important in real-life implementations of such methods - this point is discussed next (under *Computational considerations*).

Overall, even though classification improves with larger document window sizes, it is interesting to see that adequately good predictions can be achieved with relatively few recent documents representing each student. This is also seen in Figures 4 and 5, which show that folds 0, 1 and 4 seem to contain either global or local maxima at around 4-8 documents per student (three out of five data folds correspond roughly to 18 out of 30 students). The line in these figures shows the number of documents tested. The number drops for each bar going from 1 to 18 as discussed above (i.e. because a document is only tested if there are at least x older documents by the same student to put in the training set.)

Note that the above results may be affected by the difference in writing style between high- and low-grade students. As can be seen in Figures 6 and 7, prediction quality varies a lot between high-grade and low-grade students. The reason for this might be that high-grade students have a more consistent writing style, but also that they produce more text per document (see Table 2). More text per document means more data per student, which tends to generally improve prediction quality.

Computational considerations. Using the five most recent documents of a student for authorship attribution cuts down computation time notably, compared to using all his documents. Specifically, the SVM execution with a linear kernel and using the five most recent documents per student on the full dataset requires a total 72.7 seconds for model creation and prediction (on a Intel Core i7-3520M CPU with two 2.90GHz processors and multi-threading). The creation and classification was parallelised yielding almost 100% CPU utilisation. In total, 375 tests were done, meaning an approximated average CPU time of 72.7 x 4 / 375 = 0.78 seconds to test one document against the other 30. The time is multiplied by four because of the dual cores and the multi-threading, which might be an overestimation. In comparison, using all documents per student (16 documents per student on average), implies testing 1 document against 96 others, and required 68.4 seconds. Only 51 tests were done when using all 16

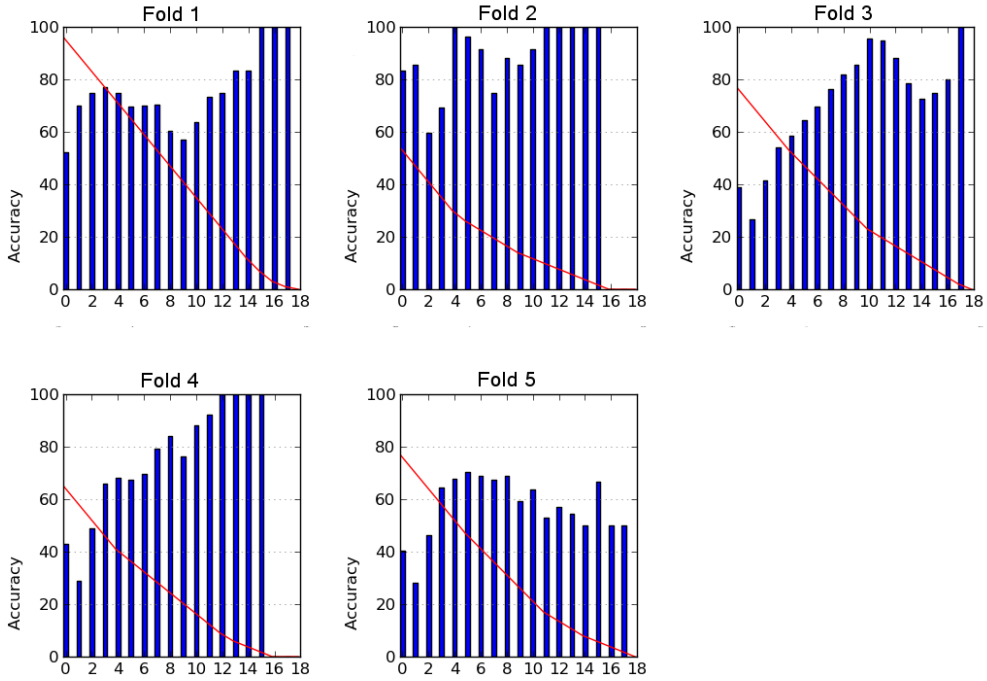

Fig. 4. Classification accuracy (y axis) versus number of documents used for classification (x axis) per fold. The documents on the x axis are sorted chronologically from the most recent at position 0 to the least recent at position 18 or lower (not all students have the exact same amount of documents in their profile). The line indicates the number of documents tested. The top 1000 most common n-grams are used for classification.

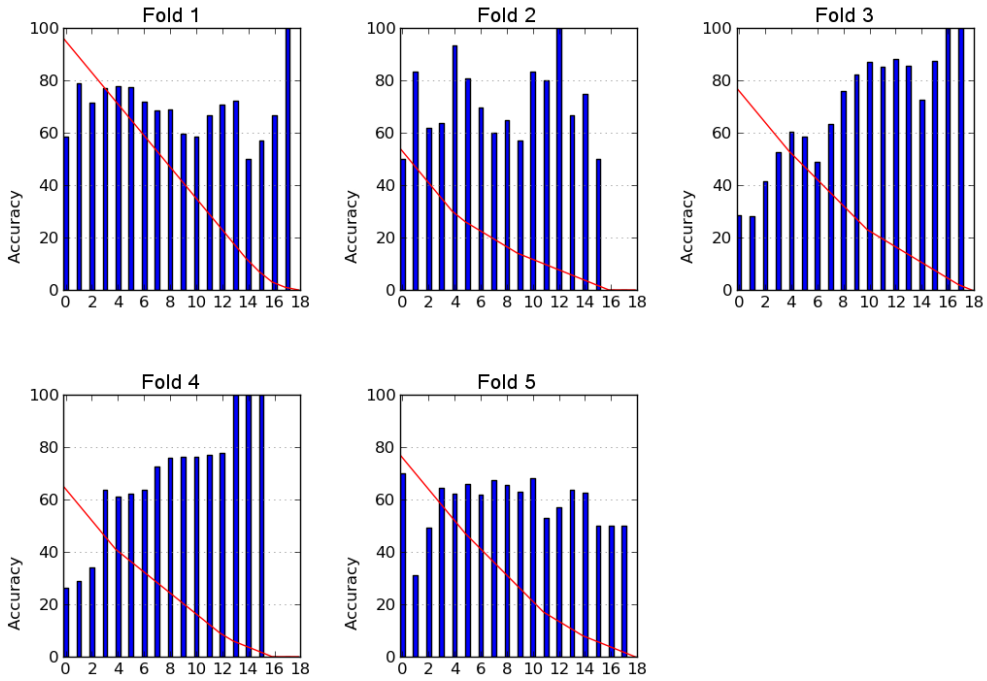

Fig. 5. Same as in Figure 4, but here only the top 500-1500 most common n-grams are used for classification.

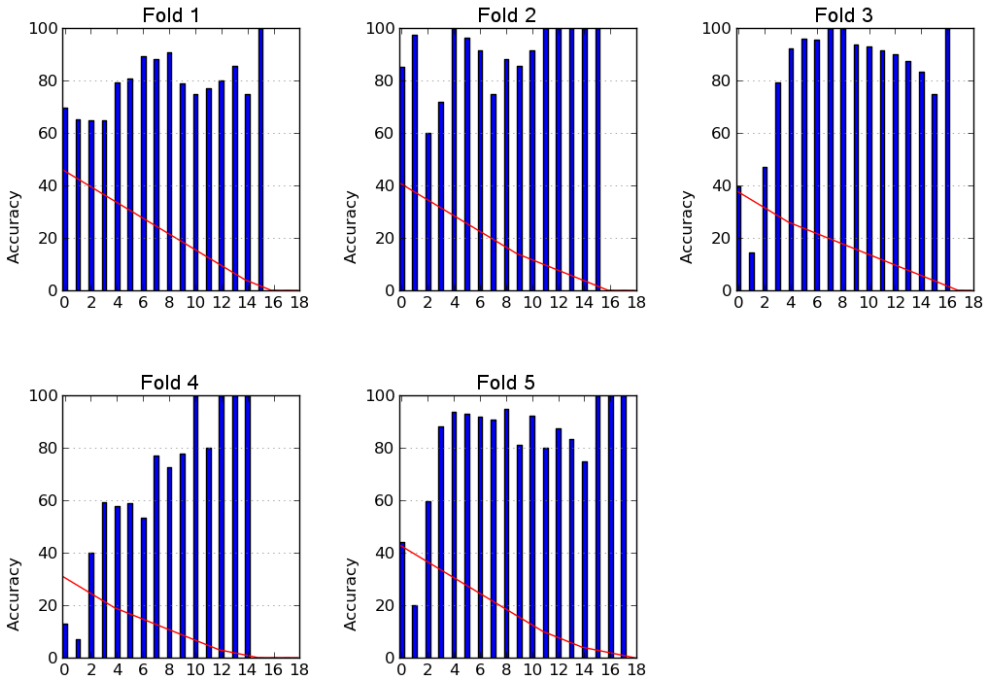

Fig. 6. Same as in Figure 4, but here only for high-grade students (15 out of 30 students).

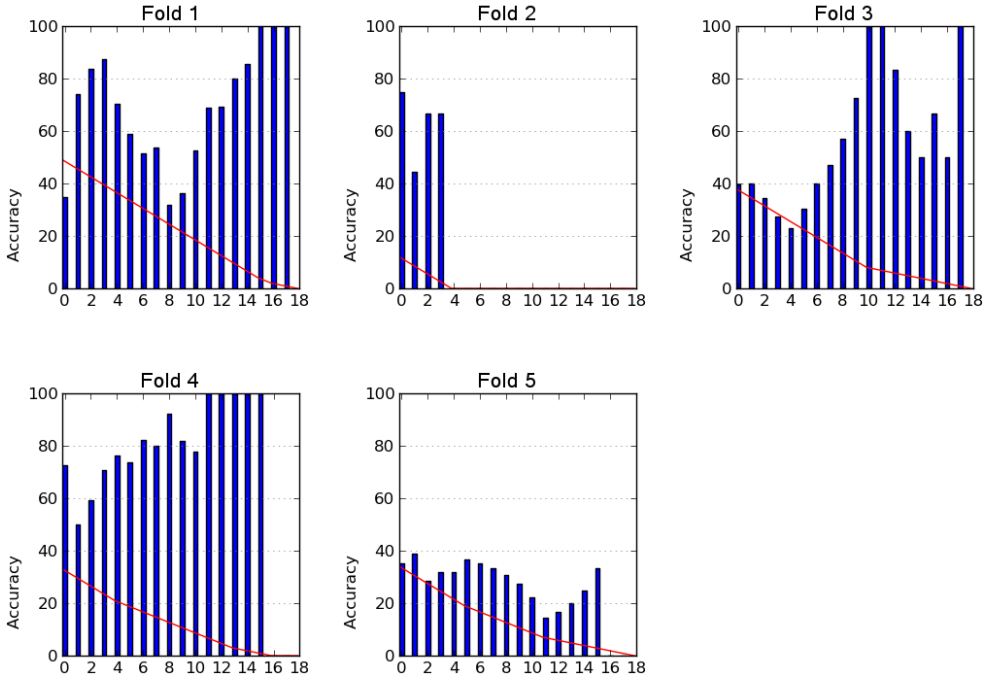

Fig. 7. Same as in Figure 4, but here only for low-grade students (15 out of 30 students).

documents, resulting in an average time per test of 68.4 x 4 / 51 = 5.4 seconds. With 16 documents, the amount of training 51 test data increased by 320%, but the computation time increased by 692%. The exact computational complexity of the LIBSVM SVM version is not known, but it is according to Hsu et al. [4] not linear, as the results above also indicate.

Practically the above indicates that using fewer, more recent documents for authorship attribution, as opposed to the full student profile, might be a feasible and computationally attractive alternative, especially for scenarios requiring quick response and increased computations. One such occasion is the final secondary school written exams, where a very large number of authors (in the range of thousands) submit documents at the same time, teachers have a significantly increased workload with a relatively quick deadline for submitting their assessments, and the online submission handling system needs to respond efficiently.

The above computational considerations imply that, with our current setup, using all the data in the Lectio database (>100,000 student profiles) is too computationally demanding. One way of addressing this is inspired by recent work by Koppel et al. [10], who propose using so-called data subset impostors, instead of testing against a possibly very large data set. Such methods would potentially reduce the amount of data needed to make accurate predictions, with considerable impact upon the usability of this type of classification for real-life systems. Taking for example the final written Danish exam for secondary school students, we reason as follows: in 2012 140,960 students were enrolled at a Danish secondary school[9]; assuming that students are evenly distributed across the three years of study, roughly 47,000 students will attend the final written Danish exam at the same time. Using the execution time found above, and using the five most recent documents per student, the total CPU time needed would be $\frac{47000 \times 0.78}{3600} = 10.2$ hours (using our limited computational resources!). Since each test is completely independent of the others, the wall-clock time should be linearly reducible with the amount of CPUs used.

5 Conclusions

We looked at the problem of authorship attribution in the domain of plagiarism detection for secondary education, as part of digital learning systems that handle student - teacher interaction, e.g. in the form of assignments, examinations, projects, feedback, and so on. We focussed on temporal context of the students, i.e. how their change in writing style during the last three years of their secondary education may impact the accuracy of authorship attribution systems. We used real-life data provided by the largest digital learning platform providers in Denmark, MaCom, and sampled 30 student profiles of both high-grade and low-grade students. While using all the documents in a student profile yielded a classification accuracy of 84%, interestingly, reducing the number of documents used to the five most recent yielded a classification accuracy of 71.9%. This drop

[9] http://www.statistikbanken.dk/

in accuracy is not detrimental, given that (i) these systems aim to assist teachers, not replace them in their decisions, hence manual checks by teachers are always applied; (ii) the computational gains brought in by using fewer recent documents per student are significant and practically mean that the corresponding systems can handle data-intensive cross-country examinations more efficiently. This consideration is not to be ignored: for instance, in Denmark, in 2011, the system handling all secondary education end-of-year examinations failed, with dire practical consequences to the teachers' workload but also the wider societal trust[10].

There are several caveats to this study that could improve in the future. For instance, improvements to the classification accuracy reported in this work can come from: (1) varying the order of n-grams to generate the SVM feature vectors; (2) experimenting with a more even distribution of student grades, instead of the extremity of the grading scale analysed in this work; (3) using a linear as opposed to general purpose SVM classifier, since the linear SVM seems to perform best and since classifiers only doing linear classification exist and are much faster [4]; (4) making predictions on smaller document segments, as opposed to whole documents, aiming to identify for instance plagiarised quotations. Identifying and removing quotations could possibly generate more accurate authorial fingerprints; (5) experimenting with a bigger variety of feature sets (e.g. based on term frequencies) and also applying them in a cascaded fashion (first the primary, then the secondary) as described in Section 2. Furthermore, the problem of skewness of the classes when doing SVM classification might be avoided by changing from a one-against-all approach used here, to a one-against-one approach. Practically this means that each student is tested against all the other students in turn. If the student document is classified correctly in all tests, then it is marked as the student's own work. Finally, it worth looking into modelling the student's personal progression in writing style. When generating an authorial fingerprint from one document, it might be the case that the specific document deviates a bit from the general writing pattern of the student. Removing these fluctuations or enhancing a general pattern could be useful. If the feature vector of the SVM is viewed as a signal, methods from signal processing might be used for improving prediction quality, for example amplifying or smoothing of the feature vector.

Acknowledgments We thank MaCom for giving us access to the data and the Lektio system, and for supporting this project.

References

1. A. Bugarin, M. Carreira, M. Lama, and X. M. Pardo. Plagiarism detection using software tools: a study in a computer science degree. In *2008 European University Information Systems Conference, Aarhus, Denmark*, pages 72.1–72.5, 2008.

[10] http://newz.dk/skoletest-igen-ramt-af-nedbrud

2. W. R. de Oliveira, E. J. R. Justino, and L. S. Oliveira. Authorship attribution of electronic documents comparing the use of normalized compression distance and support vector machine in authorship attribution. In *Proceedings of the 19th International Conference on Neural Information Processing - Volume Part I*, ICONIP'12, pages 632–639, 2012.
3. M. Farach, M. Noordewier, S. Savari, L. Shepp, A. Wyner, and J. Ziv. On the entropy of DNA: Algorithms and measurements based on memory and rapid convergence. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, pages 48–57, 1995.
4. C.-W. Hsu, C.-C. Chang, and C.-J. Lin. *LIBSVM: A Practical Guide to Support Vector Classification*. http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf, 2003.
5. P. Ingwersen. A context-driven integrated framework for research on interactive IR. *Document, Information & Knowledge*, 126(6):44–50 (in Chinese version) and 11 (in English version), 2008.
6. P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
7. F. Iqbal, R. Hadjidj, B. C. M. Fung, and M. Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Digit. Investig.*, 5:S42–S51, Sept. 2008.
8. P. Juola. Authorship attribution. *Found. Trends Inf. Retr.*, 1(3):233–334, Dec. 2006.
9. M. Koppel, J. Schler, and S. Argamon. Authorship attribution in the wild. *Lang. Resour. Eval.*, 45(1):83–94, Mar. 2011.
10. M. Koppel, J. Schler, S. Argamon, and Y. Winter. The fundamental problem of authorship attribution. *English Studies*, 93(3):284–291, 2012.
11. O. V. Kukushkina, A. A. Polikarpov, and D. V. Khmelev. Using literal and grammatical statistics for authorship attribution. *Probl. Inf. Transm.*, 37(2):172–184, Apr. 2001.
12. S. Kumpulainen and K. Järvelin. Information interaction in molecular medicine: Integrated use of multiple channels. In *Proceedings of the Third Symposium on Information Interaction in Context*, IIiX '10, pages 95–104, 2010.
13. K. Luyckx and W. Daelemans. Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 513–520, 2008.
14. N. Plotkin and A. Wyner. An entropy estimator algorithm and telecommunications applications. In G. Heidbreder, editor, *Maximum Entropy and Bayesian Methods*, volume 62 of *Fundamental Theories of Physics*, pages 351–363. Springer Netherlands, 1996.
15. J. Savoy. Authorship attribution based on specific vocabulary. *ACM Trans. Inf. Syst.*, 30(2):12:1–12:30, May 2012.
16. J. Savoy. Authorship attribution based on a probabilistic topic model. *Inf. Process. Manage.*, 49(1):341–354, Jan. 2013.
17. J. Savoy. Feature selections for authorship attribution. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 939–941, New York, NY, USA, 2013. ACM.
18. Y. Seroussi, F. Bohnert, and I. Zukerman. Authorship attribution with author-aware topic models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 264–269, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

19. Y. Seroussi, I. Zukerman, and F. Bohnert. Authorship attribution with latent dirichlet allocation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 181–189, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
20. C. E. Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, 30(1):50–64, January 1951.
21. E. Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009.
22. Y. Zhao and J. Zobel. Searching with style: Authorship attribution in classic literature. In *Proceedings of the Thirtieth Australasian Conference on Computer Science - Volume 62*, ACSC '07, pages 59–68, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.