The Applicability of Authorship Verification to Swedish Discussion Forums

Lukas Lundmark, Fredrik Johansson, Björn Pelzer, Lisa Kaati, Johan Fernquist
Swedish Defence Research Agency (FOI)
Stockholm, Sweden
Email: lukas.lundmark@foi.se

***Abstract*—The authorship verification problem of determining whether two collections of textual content have been written by the same author or not is relevant in several contexts, e.g., when law enforcement officers try to find out whether a suspect with a known user account has other user accounts in the same or other web forums. In this paper, we evaluate how well the recently suggested attention-based hierarchical neural network approach ADHOMINEM works and if it can be used to link user accounts on Swedish discussion forums. The results are encouraging and show that using ADHOMINEM is a promising way forward when linking user accounts both on the same discussion forum and in cross-domain settings in which users write on a large variety of topics.**

I. INTRODUCTION

Law enforcement analysts and investigators are often concerned with problems involving digital messages or social media user accounts on various platforms. Real-world use cases may involve having to analyze who (if any) out of a pool of suspects have written a threatening anonymous e-mail to a politician, or whether the identity of an unknown drug seller on an illegal marketplace can be revealed by linking forum posts to less anonymous accounts on other social media platforms.

This can be achieved by identifying individual linguistic features such as characteristics of spelling, grammar, and stylistic choices in a text and then find other texts that have the same linguistic features. Such linguistic analysis is usually seen as a part of forensic linguistics [18]. Although there have been several occasions where linguistic analysis has been used as evidence in court, manual analysis of linguistic features is far from perfect. Moreover, there are limits to how much data can be managed using manual analysis. For such reasons, much research has been devoted to computer-based authorship analysis.

A particular linguistic analysis task is *authorship verification* (AV), which can be defined as the problem of comparing the style of two (sets of) documents in order to decide whether they are likely to have been written by the same author or not. Several different approaches have been suggested for authorship verification. Many of these rely on extracting various stylometric features that are used as input to (shallow) machine learning-based classifiers. A large variety of stylometric features have been proposed in the literature, including various lexical, syntactic, and semantic features. Hand-crafting such features has the advantage that human experts can decide on which properties to take into account when attempting to compare whether two (sets of) documents have been written by the same author or not. Such methods work reasonably well but are restricted to the stylometric features experts identify as relevant. Progress in deep learning has led to a revolution in natural language processing. Deep learning-based methods have the ability to learn useful representations automatically, which can be seen as a type of automatic feature extraction. Lately, interesting deep learning-based approaches have been suggested as alternatives to more traditional approaches to authorship verification.

In this paper, we utilize a deep learning-based technique known as ADHOMINEM [2] for authorship verification. ADHOMINEM has previously been investigated for use cases that involve deciding whether two reviews in English have been produced by the same author.

To the best of our knowledge, we are the first to use this technique to determine whether user accounts on different web forums belong to the same person. Moreover, this research is carried out for users writing in Swedish, which is a language that this kind of hierarchical neural network-based authorship verification methods previously has not been used for.

Outline The rest of this paper is outlined as follows. In Section II we present an overview of related work and position our research contribution in relation to previous research. Next, in Section III we provide a brief description of ADHOMINEM. In Section IV we outline our formulation of the authorship verification task and present the different datasets used to train and test our model. In Section V we outline a number of experiments used to evaluate the model, and we present and discuss the results of the experiments in Section VI. Finally, the paper is concluded with some directions for future work in Section VII

II. RELATED WORK

Authorship verification is a variant of the more well-studied problem of authorship attribution. Given a text of unknown or disputed authorship and a set of candidate authors, authorship attribution aims to find the candidate most likely to have written the text. Research on such methods dates back a long time, with early works predating computerized methods [6], [14]. A useful review of features and various classifiers that

978-1-7281-6251-5/20/$31.00 ©2020 IEEE

have been suggested throughout the research literature can be found in [22]. Most existing work assumes a relatively limited number of candidate authors, but [15] have demonstrated that traditional shallow machine learning-based classifiers scale surprisingly well also for cases involving tens of thousands of candidate authors.

Research on authorship verification is not as exhaustive as for authorship attribution, but several approaches do exist. A fair amount of such methods have throughout the years been evaluated in various challenges as part of PAN[1]. Among authorship verification methods, we can separate between methods that make use of hand-crafted stylometric features and those that automatically extract features from the raw text. To learn discriminating features directly from text is potentially beneficial, but many such approaches have also shown to be highly topic-dependent. Moreover, such methods tend to lack an intuitive way to explain the resulting classifications to a user of an authorship verification system, limiting its practical value in forensic investigations. In recent years, several interesting deep learning-based approaches have shown promising results on the task as well, e.g. [2], [16].

Resolving authorship in social media introduces several additional challenges. Firstly, the amount of text is often prohibitively small. For example, the average Reddit post consists of 30 words or 180 characters. Similarly, the average tweet contains only 11 words or 67 characters[2]. This can be compared to the PAN 2020 Authorship Attribution challenge [10] which used 21,000 characters per author, or the reliable minimum of 10,000 word-tokens recommended in older literature [4]. Secondly, the number of users (i.e., possible candidate authors) can range from thousands to millions, even on a single platform.

Rocha et al. [19] provide a comprehensive overview of attempts at, and challenges with, authorship analysis in social media. Here they highlight the need for robust topic-independent features due to the small size of individual posts and the wide range of topics different user posts can cover. Sapkota et al. [20] perform authorship attribution on text compiled from different online media (such as blogs and email) with commonly used lexical and stylometric features. De Vel et al. [5], [24] exploit greetings phrases and signatures in email for authorship analysis, which to some limited degree also can be applied to social media. The ubiquity of social media data lends itself to using machine learning methods, and in particular deep learning, which can leverage a large number of training examples. The aforementioned work by Boenninghoff et al. [2] is an example of this, although their experiments focus on user reviews of products, an area of social media that is arguably atypical for the discussions commonly associated with forums and social networks.

[1]PAN is a series of scientific events / shared tasks on digital text forensics hosted annually by Webis Group.: https://pan.webis.de/

[2]Based on random samples of 1 million posts each for Reddit and Twitter.

III. ADHOMINEM

We have employed a deep learning-based method called ADHOMINEM [2] for authorship verification. ADHOMINEM is a Siamese network, i.e., a neural network architecture in which a single artificial neural network $f(\cdot)$ is applied to two inputs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in order to compare the output vectors.

The neural network backbone in ADHOMINEM consists of a Hierarchical Attention Neural Network similar to [25]. The network performs encoding at three levels: character-to-word, word-to-sentence encoding and sentence-to-document encoding using a combination of convolutional filters [13], bidirectional LSTMs [8] and attention [1].

ADHOMINEM is an approach that offers several advantages. Firstly, it is language-agnostic, and it does not require any language-specific function words or POS-tags. Instead, ADHOMINEM learns its own vocabulary and uses pre-trained word embeddings in an unsupervised fashion. Secondly, it learns a unified pseudo-metric, i.e., a generalization of the metric space where two distinct points x and y can have a distance of zero between documents, which offers more explanatory power compared to black-box classifiers. Thirdly, by combining the linguistic document embedding of the model with its attention scores, there are multiple possibilities to explain the classification to the human operator. These linguistic embeddings can, for example, be visualized using methods like t-SNE [23], such that an operator can inspect how different documents relate to each other in the embedding space. The attention scores, normally used internally by the network to determine relations between tokens, can be leveraged in ways like highlighting those tokens/sentences which the network deems to have linguistic or stylometric importance. More details about ADHOMINEM can be found in [2].

A. Authorship verification with ADHOMINEM

ADHOMINEM uses a double threshold setup which offers two disparate approaches for authorship verification: *confident verification*, where the model only predicts authorship on examples the model is sufficiently confident in, or *soft-threshold verification*, where the model predicts authorship for all examples regardless of confidence.

More formally, the model employs two fixed threshold-values: a lower threshold τ_s (same author) and an upper threshold τ_d (different authors), with $\tau_s < \tau_d$. These thresholds are set before the network is trained.

During authorship verification, given two documents D_1 and D_2, we first compute the distance between their feature-representations using the Euclidean distance,

$$d\left(\boldsymbol{y}_1, \boldsymbol{y}_2\right) = \left\|\boldsymbol{y}_1 - \boldsymbol{y}_2\right\| \tag{1}$$

where $\boldsymbol{y}_i = f(D_i)$ are the real-valued feature vectors of the documents D_i, outputted by the hierarchical attention network $f(\cdot)$.

Given the distance $d\left(\boldsymbol{y}_1, \boldsymbol{y}_2\right)$, we then use the following classification criterion for *confident authorship verification*:

$$\begin{cases} \hat{a} = 1 & \text{if } d\left(\boldsymbol{y}_i, \boldsymbol{y}_j\right) \leq \tau_s \\ \hat{a} = 0 & \text{if } d\left(\boldsymbol{y}_i, \boldsymbol{y}_j\right) \geq \tau_d \end{cases} \tag{2}$$

where $\hat{a} = 1$ means the model predicts that D_i and D_j are written by the same author, while $\hat{a} = 0$ means the model predicts that the documents are written by different authors. Document pairs (D_i, D_j) with $\tau_s < d(\boldsymbol{y}_i, \boldsymbol{y}_j) < \tau_d$ are simply ignored.

For *soft-threshold verification* we instead use the following criterion:

$$\begin{cases} \hat{a} = 1 & \text{if } d(\boldsymbol{y}_1, \boldsymbol{y}_2) \leq \frac{\tau_s + \tau_d}{2} \\ \hat{a} = 0 & \text{if } d(\boldsymbol{y}_1, \boldsymbol{y}_2) \geq \frac{\tau_s + \tau_d}{2} \end{cases} \tag{3}$$

.

In our experiments, we evaluate both of these verification approaches, although we focus primarily on confident verification since this offers more reliable predictions, at the cost of ignoring more examples.

IV. A Model for Authorship Verification on Swedish Web Forums

A. Reliable Authorship Verification with Forum Data

Authorship verification for forum and social media posts has two main issues to tackle. First, unlike many other forms of written communication, e.g., books, blogs, and emails, forum posts tend to be unstructured, generally being more akin to casual conversations. This means we cannot (reliably) exploit any formatting-related features for authorship verification, e.g., greetings or paragraph length (and even if we could, these might not generalize well across forums). Therefore, to guarantee good generalization, we are forced to focus solely on the textual content of the posts. Secondly, the amount of text in any single forum post is often extremely short, e.g., one or two sentences. Although contemporary results on single post authorship verification, such as Schwartz et al. [21], are impressive when considering the minuscule amount of text they use, their fairly low reliability makes it difficult to use their predictions as evidence, for example, in a court of law.

We instead choose to focus on combining sets of posts into what we call *profile documents*, which are then used in authorship verification. Similar approaches where multiple bodies of text are combined into profile documents have been employed by, e.g., [9] for essays.

Our approach to authorship verification is, therefore, as follows: We define an authorship verification example as a pair of two profile documents D_i and D_j from two unknown authors u_i and u_j, where both documents are compiled from disjoint sets of forum posts. The tasks is then to determine if $u_i = u_j$.

1) Document Length: Composing the profile documents from multiple, very short forum posts (the median post length is 46 word-tokens) offers an advantage over other authorship verification settings since that we can control the document size by adjusting the number of posts.

Although document length is a factor that is well established as having a major influence on the performance of authorship verification systems, few researchers explicitly try to control for document length in their experiments. In our experiments, we will explicitly control the document size and evaluate how differences in size affect the performance. In that way, we can determine suitable heuristics for the amount of text required to conduct reliable authorship verification.

When controlling for document size, the naive approach would be to use an equal number of posts for each profile document. However, since posts may vary in size, both between users and different forums, this could lead us to overestimate performance on, e.g., more verbose forums while underestimating performance on less verbose forums. We control the document size by limiting each profile document to a fixed number of sentences n. The median sentence length in our training forum is 16 word-tokens, and the median number of sentences per user is 32 sentences. We, therefore, set the default size of training documents to $n = 30$ (ca. 492 tokens per document) such that we, in theory, could construct a profile document for more than half of the users in our training set. Similar document sizes have also been used in several related works, e.g., [2], [3], [20].

We evaluate the trained model using various settings for n to get a good estimate of the document size the model requires to retain good accuracy.

2) Topical and Temporal Constraints: We also constrain the temporal span of the posts in each profile document. Previous research [26] has shown that compiling profile documents by sampling posts at random makes the verification problem significantly easier, compared to when there is no temporal overlap between posts. Similarly, we also constrain profile documents to cover posts from only a single topic in order to further increase the difficulty of the task. We argue that profile documents of this type – i.e., consisting of a small set of posts that were posted in a limited time frame and only covering a limited range of subjects – represent a likely real-world scenario in which a forensic linguistic investigation could be needed.

B. Generating Data

1) Data Gathering: To train our model, we first construct a large authorship verification dataset from one of Sweden's most popular discussion forums (which we denote as DF1). The discussion forum is pseudonymous, i.e., users are identified via unique user handles but are otherwise anonymous. The forum consists of a number of unique main boards, each covering a unique topic such as IT and economics. We regard the main board a post was posted under as the topic of that post.

2) Compiling Profile Documents: For each user, we separate the user's forum posts based on topic. For each group of topic posts, we then sort the posts in ascending order based on the date and time of posting. The sorted posts are then concatenated into a single large document, which is then split into a set of smaller, non-overlapping documents, with each smaller document consisting of exactly $n = 30$ sentences. We use these 30-sentence documents as profile documents during both training and testing.

We discard users who could not produce at least two profile documents on at least two different topics. This reduces the

number of users in the dataset from roughly 460,000 to 77,256. Note that users still can have more than four profile documents.

3) Generating AV Examples: To get balanced training data, we generate an equal number of positive and negative example pairs. Keeping in line with the notation of [3] we denote positive samples as $a = 1$ and negative samples as $a = 0$. We also generate an equal number of intra- and inter-topic pairs (i.e., where the two documents cover the same or different topics, respectively). We denote intra-topic examples as $c = 1$ and inter-topic examples as $c = 0$. In each training epoch, we create four example pairs for each user u: $(a = 0, c = 0)$, $(a = 0, c = 1)$, $(a = 1, c = 0)$, $(a = 1, c = 1)$, where each pair consists of two disparate profile documents. Here, for $a = 0$, we randomly sample a different user, and for $c = 0$, we randomly sample a different topic. After each epoch we augment the data by resampling these examples pairs for each user.

C. Training the Model

We use the dataset described above to train the model. The user set from DF1 $\mathcal{U}$ is split into three disjoint sets: $\mathcal{U}_{\text{train}}$, $\mathcal{U}_{\text{validation}}$ and $\mathcal{U}_{\text{test}}$, consisting of 80%, 10% and 10% of the original set of users respectively. The ADHOMINEM implementation is trained using example pairs from the set of example documents $\mathcal{D}_{\text{train}}$ for the train-user set $\mathcal{U}_{\text{train}}$. Pairs are sampled as outlined in Section IV-B, resulting in 247,232 training pairs in each epoch.

Early stopping is used during training to prevent overfitting: the model trains until the loss on the validation document set $\mathcal{D}_{\text{validation}}$ has not improved for three consecutive training epochs, whereupon the best model is selected.

In the experiments we use the remaining unseen set of test-users $\mathcal{U}_{\text{test}}$ and their example documents $\mathcal{D}_{\text{test}}$.

1) Tokenization and Sentence Boundaries: For document tokenization, we employ the natural language toolkit NLTK[3]. We use the *Punkt* tokenizer [12] for Swedish to detect sentence boundaries and use a regex-based word-tokenizer to split the sentences into word-tokens.

2) Vocabulary and Embeddings: We construct a Swedish word-level vocabulary by first converting all documents to lower case. We collect all unique tokens $\{w|\forall w \in \mathcal{D}_{\text{train}}\}$ in the training documents $\mathcal{D}_{\text{train}}$ and then discard any token that occurred less than 20 times in $\mathcal{D}_{\text{train}}$. We use a catch-all token <UNK> to replace these missing tokens. Removing these rare tokens reduces the vocabulary size from 5,357,243 down to 267,899. Hiding rare tokens not only reduces vocabulary size but also doubles as a form of topic-masking.

As word embeddings $\boldsymbol{x}^{(w)}$ we use pre-trained CBOW word-embeddings[4] for Swedish [7]. We denote the vocabulary of the pretrained embeddings as $\mathcal{V}_{\text{pre}}$ and the vocabulary derived from $\mathcal{D}_{\text{train}}$ as $\mathcal{V}_{\text{train}}$. Before training we first discard all pre-trained token-embeddings for all tokens w where $w \notin (\mathcal{V}_{\text{pre}} - \mathcal{V}_{\text{train}})$. We then randomly initialize embeddings for all tokens w where $w \notin (\mathcal{V}_{\text{train}} - \mathcal{V}_{\text{pre}})$. All word-embeddings are then updated jointly with the network during training.

[3]https://www.nltk.org/

[4]https://fasttext.cc/docs/en/crawl-vectors.html

TABLE I
THE THREE DIFFERENT DATASETS USED FOR TRAINING AND TESTING.

Dataset	Source	# authors	# topics
Intra-Domain (test)	DF1	7726	16
Domain Shift	DF2	8755	17
Cross-Domain	DF1 and DF3	969	2

The character vocabulary is constructed by first collecting all unique characters, upper and lower-case, in all training documents $\{c|\forall c \in \mathcal{D}_{\text{train}}\}$. All characters that occurred less than 100 times in $\mathcal{D}_{\text{train}}$ are then discarded, reducing the character vocabulary count from 3,442 to 366. The character-embedding size is set to $d^c = 10$. The embeddings are then randomly initialized, and updated jointly with the network during training.

3) Implementation details: We implement ADHOMINEM using PyTorch 1.5 [17]. We set $\tau_s = 1$ and $\tau_d = 3$. Like [3], the output size of the word-to-sentence LSTM encoders are set to $d^s = 150$, and the output size of the sentence-to-document encoders to $d^d = 75$. However, since ADHOMINEM [2] instead uses bidirectional LSTMs, the outputed sentence embeddings have an actual size of $2 \cdot d^s$, and the document-embeddings a size of $2 \cdot d^d$. The output size of the final linear layer is set to $d^y = 30$. The network is trained using an Adam [11] optimizer with a cyclic learning rate schedule. Dropout is applied with a keep probability of $p = 0.9$ to the convolutional filters, the attention weights, and the final metric projection. We use a batch size of 32 and train using two NVIDIA GeForce RTX 2080.

V. EXPERIMENTS AND SETUPS

A. Evaluation Settings

We evaluate our ADHOMINEM authorship verification model using three different datasets: Intra-Domain, Domain Shift, and Cross-Domain. The three different datasets contain data from three different discussion forums: DF1, DF2, and DF3. The datasets are listed in Table I, and the three different evaluation settings are described in more detail below.

Evaluation Setting 1: Intra-Domain
For the first evaluation setting, we employ the most common setup employed in authorship analysis research: by using an unseen test user set from the same domain as the training data. More specifically, we use the test user set $\mathcal{U}_{\text{test}}$, outlined in Section IV-C.

We consider this setting as an idealistic special case, rarely encountered when applying AV in practice. Therefore, we use this setting more as a way to highlight how much we can overestimate performance when compared to less idealistic settings, rather than a way to evaluate actual performance.

Evaluation Setting 2: Domain Shift
In the second setting, we evaluate the model in a more realistic scenario. We apply the model not only to unseen users but to users collected from an entirely different domain/discussion

forum. We consider this to be a more general (and realistic) version of evaluation setting 1, and we use it to showcase if we can feasibly apply authorship verification in forums that do not have sufficient users to train a model.

Here, we compile data from a different discussion forum that we denote as DF2. DF2 has a much more limited user-base compared to DF1, and it has a disparate target audience. Similar to DF1, DF2 consists of multiple different main-boards, with each board focusing on a specific topic. Here, however, the topic-range is much narrower than in DF1, with topics focusing primarily on child-rearing and other aspects of domestic life.

Evaluation Setting 3: Cross-Domain

In the third setting, we try to generalize the setting even further. Previously, we have assumed the documents in the pair we want to verify are from the same domain. Here, we instead verify documents from disparate discussion forums.

We compile a cross-domain dataset that consists of document pairs spanning two different discussion forums: our original forum DF1 and a discussion forum DF3 that have a focus on IT and technology. To find accounts that are created by the same user in both forums, we first identify all the usernames in $\mathcal{U}_{\text{test}}$ that exist in both DF1 and DF3. Two accounts on two different forums having identical usernames do not necessarily mean the accounts correspond to the same real-life user. Therefore, to minimize the risk of incorrect labeling, we limit our set of usernames to only "unique" usernames, i.e., names that are unlikely to have been chosen by two different people. For this purpose, we define a "unique" username as:

- Consisting of more than four characters.
- Not being a standard Swedish or English Name (first or surname).
- Not being a reference to famous cultural icons (e.g., Spongebob or Sonic the Hedgehog).
- Not being an overly generic phrase.

We tasked three human annotators to tag usernames as being unique or not based on these criteria. Only users that were labeled as unique by all three annotators were kept. We assume that these usernames represent the same people posting in both forums.

The profile documents are compiled by sorting all user posts from the shared usernames in either forum in adjacent order based on posting time. The texts are then concatenated and split into sub-documents, each consisting of 30 sentences. Users who have posted less than 30 sentences in either forum are excluded. The resulting dataset consists of 969 unique users. Unlike previous settings we only generate two example pairs per user at each epoch: $(a = 1, c = 0)$ and $(a = 0, c = 0)$. Here, $c = 0$ also denotes disparate forums, rather than disparate topics.

B. Experiments

Experiment 1: Accuracy in Different Settings

To get an insight into how different types of data affect the performance of the model, we investigate how the accuracy changes for the three different settings. In each setting, we use profile documents with a size of $n = 30$ sentences.

In our experiments, we focus on two aspects. First, we investigate the soft-threshold authorship verification accuracy of the model and how it is affected by changes in domain setting. Secondly, we use the double threshold setup of ADHOMINEM and investigate how different settings affect the verification accuracy of confident prediction, as well as the proportion of examples that can be predicted with confidence.

Experiment 2: Impact of Document Length

In the second experiment, we investigate in detail how the length of the input-documents affects the accuracy metrics mentioned above.

Here we perform the same experiments as in Experiment 1, but use documents of increasing sizes, from $n = 10$ to $n = 100$ in increments of 10. We first select the subset of users from each dataset that can produce at least two 100-sentence profile documents to make sure we always evaluate on the same settings. This reduces the number of unique users to 3,398 for Intra-Domain, 3,398 for Domain Shift, and 637 for Cross-Domain.

In order to make sure that we do not overestimate the performance by selecting a much easier set of users, we first evaluate the model on $n = 30$ documents and find the average accuracy to be, in fact, marginally worse when compared to the full datasets. We are, therefore, confident that we are not overestimating the performance.

VI. RESULTS

Experiment 1: Accuracy in Different Settings

The results for Experiment 1 is showed in Table II and Table III. As can be noted in Table II, ADHOMINEM performs well on our Swedish forum post authorship verification, with an average soft-threshold accuracy of 81% on quite small documents. The results also show that the overall verification accuracy is reasonably consistent when the model is applied to unseen forums and even when performing authorship verification over different forums.

Compared to Intra-Domain, Domain Shift has a significant disparity between the soft-threshold accuracies for positive and negative examples (see Table II). The same holds true for intra- and inter-topic examples.

[5] $\forall l = (a, c)$ denotes all four possible label combinations.

TABLE II
SOFT-THRESHOLD ACCURACY FOR EACH LABEL CONFIGURATION IN ALL THREE SETTINGS ($n = 30$)

Labels	Intra-Domain acc (%)	Domain-Shift acc (%)	Cross-Domain acc (%)
$\forall l = (a, c)$[5]	81.1	79.9	70.6
$a = 0, c = 0$	81.8	74.6	79.1
$a = 0, c = 1$	79.7	71.2	-
$a = 1, c = 0$	79.5	84.2	62.1
$a = 1, c = 1$	83.3	89.5	-

TABLE III
ACCURACY WHEN $d(\cdot,\cdot) \leq \tau_s$ OR $d(\cdot,\cdot) \geq \tau_d$ FOR EACH LABEL CONFIGURATION IN ALL THREE SETTINGS ($n = 30$).

Labels	Intra-Domain		Domain Shift		Cross-Domain	
	acc (%)	# (%)	acc (%)	# (%)	acc (%)	# (%)
$\forall l = (a, c)$	95.0	15.1	95.5	13.5	85.6	12.6
$a = 0, c = 0$	95.2	20.7	96.6	14.1	80.3	18.7
$a = 0, c = 1$	96.1	18.9	97.7	13.1	-	-
$a = 1, c = 0$	94.2	9.5	93.9	11.9	90.9	6.5
$a = 1, c = 1$	94.0	11.2	93.8	14.9	-	-

We hypothesize that this is due to a disparity in the distribution of *linguistic markers* in the posts; possibly an absence of certain markers which have high significance for verifying authorship in DF1. This results in document vectors $\boldsymbol{y}$ by default being closer to each other in the embedding space, in turn causing a bias toward positive classifications. This bias can also be observed in the comparatively larger fraction of positive samples that are verified with confidence and their slightly lower accuracy in Table III.

The accuracy for the confident verifications (confident accuracy) that is shown in Table III is more promising. The accuracy is very stable; it is even slightly better between both Intra-Domain and Domain Shift. This highlights one of the strengths of the double threshold approach: that it can retain a high, robust accuracy in more challenging settings, at the cost of not verifying a subset of the examples.

In the Cross-Domain setting, we observe a noticeable drop in the soft-threshold accuracy, as well as in the confident accuracy on negative examples. However, the confident accuracy remains consistent for positive examples, which is arguably more important since we want to avoid false positives, especially in a system employed by, e.g., law enforcement.

We note that this setting has the opposite problem to that of the Domain Shift setting. Here, the model is biased towards negative predictions, judging by the small fraction of confident, positive classifications, and lower confident accuracy for negative samples.

This bias towards negative classifications is not entirely surprising. Assuming that our hypothesis regarding disparate linguistic marker distribution in different domains holds true, the expected difference between document vectors will be larger in this setting, thereby causing more negative predictions. Here, unlike the Domain Shift setting, the double threshold approach is unable to compensate for this disparity, at least for the negative samples. Future research will focus on methods to make the model more robust to these forms of domain disparities.

Experiment 2: Impact of Document Length

In Figure 2 the soft-threshold accuracy for varying document sizes is shown. Unsurprisingly, we see improvements in the accuracy in all settings when increasing the document length. The same thing holds for the confident accuracies (Figure 3). Similarly, Figure 1 shows that the model gradually increases the fraction of confident examples as the length of the documents increases.

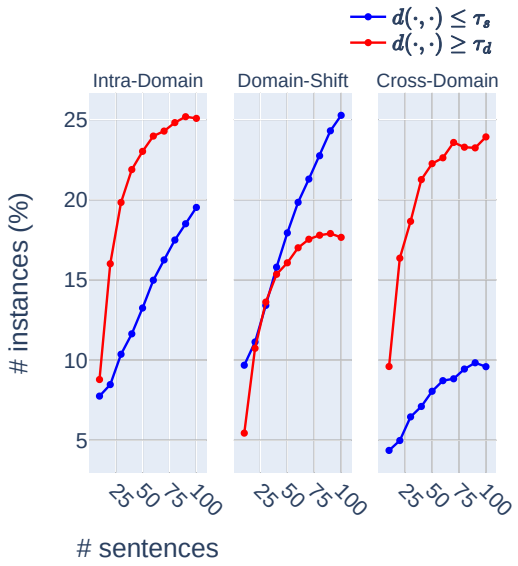

Fig. 1. Percentage of example pairs that are verified with confidence for varying document lengths.

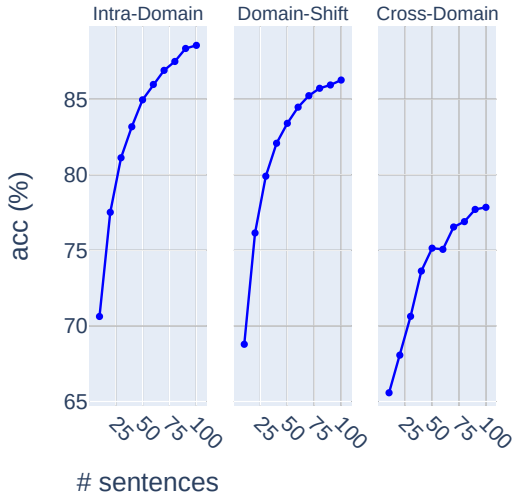

Fig. 2. Accuracy for soft-threshold verification for varying document lengths

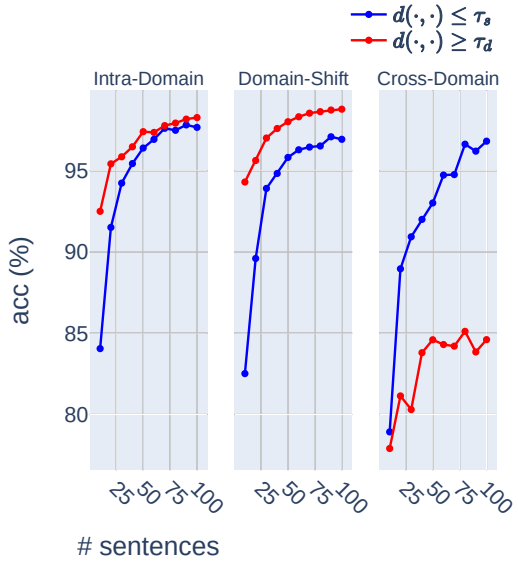

Fig. 3. Accuracy for "confident" verification for varying document lengths.

However, we start to see diminishing returns in accuracy when increasing the amount of text to more than 80-90 sentences per document. The outlier in these results is when we test our model in the Cross-Domain, which again exhibits the aforementioned issue with low confident accuracy for $d(\cdot,\cdot) > \tau_d$. More noticeably, however, the accuracy does not significantly improve when increasing the document sizes. We find $n = 60$ to be a suitable lower limit for document size to ensure good confident predictions, giving $\geq 95\%$ in all settings. However, for Intra-Domain and Domain Shift, $n = 40$ appears sufficient. We also note that all accuracies in all settings deteriorate rapidly (for e.g., $n = 10, n = 20$). This highlights that finding a suitable heuristic lower document limit is essential for a system to be reliable.

However, we do realize that there are problems with using these simple heuristics. One is that it is quite inflexible. For example, there are probably many 20-29 sentence documents that we could use for author verification, but which we now choose to discard. A more flexible approach would be to let the verification model itself determine whether a document is suitable to use for verification. In feature research, we will explore such methods, e.g., by utilizing the pre-normalized attention scores in the encoding network $f(\cdot)$ to see if the network can detect any exploitable linguistic markers.

Another interesting aspect that we want to investigate in future research is how the document size used during training affects final performance. As can be observed in the results of Experiment 2, we start to seeing diminishing returns in accuracy when using more than 80 sentences. We hypothesize that by using a fixed document size during training, we bias the model's sentence-to-document encoder, thereby making it unable to handle documents that differ too much in size from the training documents. In future research, we want to investigate how we can train our model to be stable for all document sizes, e.g., by using varying sizes for training documents or an ensemble of multiple models trained using different document sizes.

VII. Conclusions and Future Work

In this paper, we have applied the deep learning-based technique known as AdHominem [2] for authorship verification in Swedish. We have trained a model and tested the capability of the model to get an understanding of how well such a model will work in a real scenario. The results show that AdHominem generally is a good candidate for automatic authorship verification in realistic scenarios. One caveat can be raised regarding the performance of the model on smaller documents; however, this is a problematic situation for authorship verification techniques in general.

More research needs to be done to determine whether a document is suitable for automatic authorship verification or not. Ideally, the author verification system should be able to determine if a document is suitable for authorship verification automatically.

For future work, we plan to extend our method of example selection and profile documents to English language data, to allow a better comparison with other approaches and benchmarks. We also aim to find a more stable approach to verifying profile documents of varying sizes.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[2] B. Boenninghoff, S. Hessler, D. Kolossa, and R. M. Nickel. Explainable authorship verification in social media via attention-based similarity learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 36–45. IEEE, 2019.

[3] B. Boenninghoff, R. M. Nickel, S. Zeiler, and D. Kolossa. Similarity learning for authorship verification in social media. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2457–2461. IEEE, 2019.

[4] J. Burrows. All the way through: testing for authorship in different frequency strata. *Literary and Linguistic Computing*, 22(1):27–47, 2006.

[5] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *SIGMOD RECORD*, 30:55–64, 2001.

[6] A. Ellegård. *Who was Junius? A statistical method for determining authorship: the Junius letters 1769–1772*. Gothenburg Studies in English; 13. Acta Universitatis Gothoburgensis, Göteborg, 1962.

[7] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] M. Hürlimann, B. Weck, E. van den Berg, S. Suster, and M. Nissim. Glad: Groningen lightweight authorship detection. In *CLEF (Working Notes)*, 2015.

[10] M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, M. Potthast, and B. Stein. Overview of the cross-domain authorship verification task at pan 2020. In *CLEF*, 2020.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[12] T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] A. Q. Morton. *Literary Detection: How to Prove Authorship and Fraud in Literature and Documents*. Scribner, 1978.

[15] A. Narayanan, H. Paskov, N. Gong, J. Bethencourt, E. Stefanov, E. Shin, and D. Song. On the feasibility of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy (SP)*, pages 300 –314, may 2012.

[16] J. Ordoñez, R. R. Soto, and B. Y. Chen. Will longformers pan out for authorship verification. *Working Notes of CLEF*, 2020.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[18] M. Rathert and J. Visconti. *Handbook of Communication in the Legal Sphere / Jacqueline Visconti.* Handbooks of Applied Linguistics [HAL] ; 14. De Gruyter Mouton, Berlin ; Boston, 2018.

[19] A. Rocha, W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. Carvalho, and E. Stamatatos. Authorship attribution for social media forensics. *IEEE Trans. Inf. Forensics Secur.*, 12(1):5–33, 2017.

[20] U. Sapkota, T. Solorio, M. Montes, S. Bethard, and P. Rosso. Cross-topic authorship attribution: Will out-of-topic data help? In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1228–1237, Dublin, Ireland, Aug. 2014. Dublin City University and Association for Computational Linguistics.

[21] R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, 2013.

[22] E. Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009.

[23] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[24] O. D. Vel, A. Anderson, M. Corney, and G. Mohay. Multi-topic e-mail authorship attribution forensics. In *Proceedings ACM Conference on Computer Security - Workshop on Data Mining for Security Applications*, pages –. ACM, 2001.

[25] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[26] N. Zechner. *A novel approach to text classification*. PhD thesis, Umeå universitet, 2017.