ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Detecting Ghostwriters in High Schools

Magnus Stavngaard August Sørensen Stephan Lorenzen[†]
Niklas Hjuler Stephen Alstrup [*]

University of Copenhagen - Department of Computer Science
Universitetsparken 3, Copenhagen, Denmark
† Corresponding author, e-mail: lorenzen@di.ku.dk

Abstract. Students hiring *ghostwriters* to write their assignments is an increasing problem in educational institutions all over the world, with companies selling these services as a product. In this work, we develop automatic techniques with special focus on detecting such ghostwriting in high school assignments. This is done by training deep neural networks on an unprecedented large amount of data supplied by the Danish company MaCom, which covers 90% of Danish high schools. We achieve an accuracy of 0.875 and a AUC score of 0.947 on an evenly split data set.

1 Introduction

The number of Danish high school students using ghostwriters for their assignments has been rising at an alarming rate due to the emergence of several new online services, allowing students to hire others to write their assignments[1].

We consider in this paper the problem of detecting such ghostwriting, or as it is more commonly known: *authorship verification.* Authorship verification is a common task in natural language processing [2, 3, 4]: Given author α with known texts $t \in T_\alpha$ and unknown text x, determine whether α is the author of x. Often, a set of texts $\overline{T_\alpha} = T \setminus T_\alpha$ (T denoting the complete set of available texts) not written by α is also available, which can be utilized as examples of different writing styles, when training a model. Note however, that $\overline{T_\alpha}$ is unlikely to contain examples written by the true author of x, unlike in the related *authorship identification* problem, in which the task is to determine the exact author of x, given a set of candidate authors and their texts [5, 6].

In this paper, we focus on the problem in high schools. We have access to a large data set consisting of 130K Danish essays, written by more than 10K high school students[1]. Thus we have access to a lot of different authors, each with a large amount of text. We suggest a *generalizing* technique for authorship verification (as opposed to *author specific* models); using a Siamese network working at character level (an approach inspired by [5]), writing style representations are learned and compared, in order to compute the style similarity between two texts. Using the similarity measure provided by this network, x are compared to previous works $t \in T_\alpha$, and a final answer is given by a weighted combination of the individual similarities. The data used is supplied by MaCom, the company behind Lectio, the largest learning management system in Denmark.

[*] Supported by the Innovation Fund Denmark through the Danish Center for Big Data Analytics Driven Innovation (DABAI). The authors would like to thank MaCom.

[1] The data set is proprietary and not publicly available.

Many previous approaches for authorship verification/identification are based on excessive feature selection [7, 2], but neural network approaches have also been considered, for instance [3] who utilize recurrent neural networks for identification. Previous work on Danish high school essays have used author specific models for verification/identification [6], but this work is the first neural network based approach used on this data (and, to our knowledge, in this setting).

2 Method

As mentioned, we solve the authorship verification problem in two steps. First, we solve the problem of computing the writing style similarity between two texts by learning the similarity function $s : T \times T \to [0, 1]$ using a Siamese network (Section 2.1). Second, we solve the authorship verification problem for author α by combining similarities computed between the unknown text x and the known texts $t \in T_\alpha$. We consider several different ways to combine these similarities, based on their value and relevant meta data. (Section 2.2).

2.1 Network

Several different architectures are considered, using different input channels (e.g. char, word, POS-tags), and evaluated on a validation set. The architecture of our best performing network is shown in Figure 1.

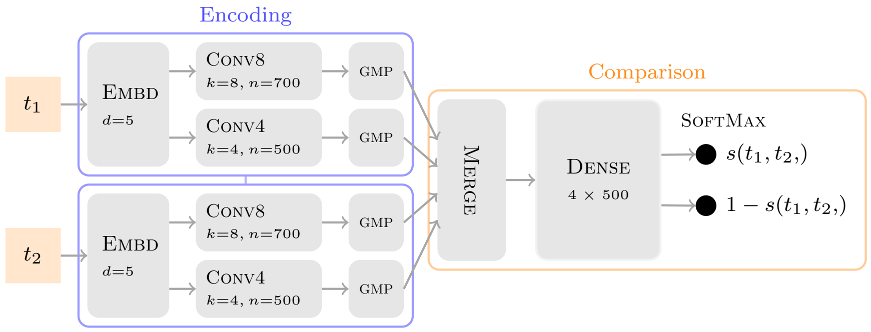

Fig. 1: Network architecture.

The Siamese network can be considered in two parts: *encoding* and *comparison*, the main idea being to learn an encoding of writing style, that the network is then able to distinguish. Our network uses only character level inputs.

The **encoding part** consists of a character embedding (EMBD), followed by two different convolutional layers: CONV8 using kernel size $k = 8$ and $n = 700$ filters, and CONV4 using $k = 4$ and $n = 500$. Each convolutional layer is followed by a global max pooling layer (GMP). The weights of EMBD and CONV8/CONV4 are shared between encoding t_1 and t_2.

In the **comparison part**, we first compute the absolute difference between the encodings in the MERGE layer. Afterwards, 4 dense layers with 500 neurons each are applied (Dense), and finally, the output is normalized by use of a softmax layer with two outputs.

2.2 Combining similarities

Having a good estimate of $s(t_1, t_2)$ for any two texts, we consider different ways to combine these similarities, in order to give the final answer to an authorship verification query. More specifically, we consider functions $C_s : \mathcal{P}(T) \times T \rightarrow [0, 1]$, such that, given x and T_α, we will answer the query positively (i.e. α is the author of x) if:

$$C_s (T_\alpha, x) \geq \delta$$

where δ is a configurable threshold, which describes how likely we are to answer positively. In the experiments, we consider several different ways to combine similarities, for instance using weighted sums, the min/max similarity or majority vote, while utilizing meta data such as time stamps and text length. From the experiments, we found that the optimal strategy was a weighted sum with weights decaying exponentially with time:

$$C_s (T_\alpha, x) = \sum_{t \in T_\alpha} e^{-\lambda \tau(t)} s(t, x) \tag{1}$$

where $\tau(t)$ denotes the time in months since t was written, and λ is a configurable parameter, which is determined experimentally.

3 Experiment

This section describes our experiments performed on the MaCom data. Section 3.1 will describe the preprocessing and partitioning of data. Baselines will be described in Section 3.2. Finally, Section 3.3 lists and discusses the final results. We use accuracy, *false accusation rate*, FAR $= \text{FN}/(\text{TN} + \text{FN})$, and *catch rate*, CR $= \text{TN}/(\text{TN} + \text{FP})$ as performance metrics.

3.1 Data

The data is partitioned into three sets: T_{train} used for training, T_{val} used for early stopping and selecting C_s, and T_{test} used only for estimating the metrics of the final models. The three sets are author disjoint, meaning no author will appear in more than one of the sets. In an effort to remove invalid data (blank hand-ins, etc.), we clean the data by filtering according to length (keeping texts with lengths between 400 and 30,000 characters). Furthermore, some texts were found to include author revealing information (such as name, address); hence we removed all proper pronouns from the texts, as well as the first 200 characters. Finally, authors with less than 5 texts were removed.

After cleaning, the data set contains a total of 131,095 Danish essays, written by 10095 authors, with an average 13.0 texts per author, and an average text length of 5894.8 characters.

For each data set, we construct two types of problem instances: SIM and AV, used for training the network and selecting the combination strategy respectively. The data set has no labelled ghostwriters, so we assume all authors to be correct[2], and construct balanced (50/50) data sets as follows:

A SIM instance simply consists of two texts t_1, t_2 and a label indicating whether the texts are by the same author. Positive samples are generated by using $t_1, t_2 \in T_\alpha$, while negative samples are generated by using $t_1 \in T_\alpha$ and $t_2 \in \overline{T_\alpha}$. An AV instance consists of a set of known texts T'_α, an unknown text x, and a label indicating whether α is (positive) or is not (negative) the author of x. Letting t_{last} denote the most recent text of T_α, samples are generated using $T'_\alpha = T_\alpha \setminus \{t_{last}\}$ with $x = t_{last}$ for the positive sample, and $x \in \overline{T_\alpha}$ chosen at random for the negative sample.

Table 1 provides an overview of the data after partitioning and preprocessing.

Data set	#authors	#texts	#SIM	#AV
T_{train}	5418	70432	934720	10836
T_{val}	989	12997	173536	1978
T_{test}	3688	47666	627744	7376

Table 1: Data set overview.

3.2 Baselines

We will compare our method to Burrows's Delta method and author specific SVMs:

Burrows's Delta method (BURROWS) [7] is a method for authorship identification based on the l_1-distance between the z-scores of word frequencies in x and in the corpus for each of the candidate authors $\beta_1, ..., \beta_k$. We adapt it for verification by sampling a set of 'wrong' authors, $\beta_2, ...\beta_k$, and querying with x and $\beta_1 = \alpha, \beta_2, ..., \beta_k$. answering positively, if x is attributed to α. The top 150 word frequencies are considered. The optimal k is determined using T_{train}.

An author specific SVM [6, 2] is trained for each author in order to recognize T_α from $\overline{T_\alpha}$. Hyper parameters and features are selected using cross validation. Forward feature selection is used, considering char, word and POS-tag n-grams for varying n. The SVM will be trained on a balanced set, meaning that only a limited amount of data is available for each SVM. However, they have previously been shown to work well in this data set [6].

3.3 Results

Methods were trained and validated on T_{train} and T_{val}. For BURROWS, we found $k = 4$ to give the best results, while the parameters $C = 10, \gamma = 10^3$ were found optimal for the RBF kernel SVM. The optimal combination strategy C_s was

[2] An undoubtedly false assumption, which will be discussed in Section 3.3

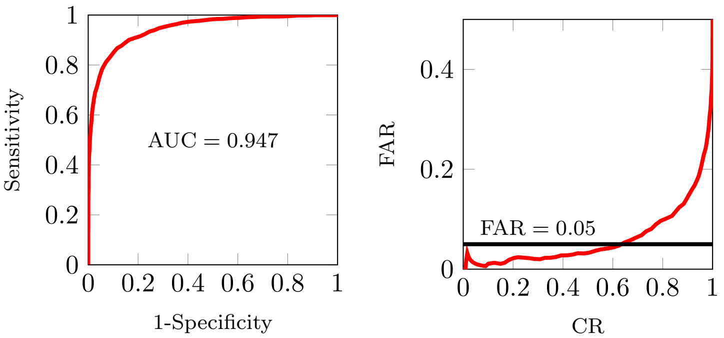

Fig. 2: ROC (left) and plot of false accusation rate/catch rate (right) on T_{test}.

found to be exponentially decaying weights (see (1)) with $\lambda = 0.1$. Furthermore, $\delta = 0.57$ was found to be optimal. Using these parameters, the baselines and our method were evaluated on T_{test}; Table 2 presents the results, while Figure 2 shows the ROC/AUC and a plot of false accusation/catch rate for our method. As it can be seen, our method clearly outperforms the baselines, on all metrics.

Method	Accuracy	FAR	CR
Burrows	0.677	0.357	0.806
SVM	0.720	0.266	0.689
Our method	0.875	0.141	0.896

Table 2: Results obtained on T_{test}

The false accusation rate is especially important considering the use case: when trying to detect ghostwriting in high schools, making false accusation can be especially devastating, as students found guilty of cheating could risk severe punishment and maybe even be expelled. Using this metric, our method performs very well, as illustrated in Figure 2 (right), a fairly low FAR can be obtained, while still catching a lot of ghostwriters. Optimizing the method on T_{val} while restricting FAR < 0.1, we achieved an accuracy of 0.864, FAR $= 0.106$ and CR $= 0.825$ on T_{test} (with exponential weighting and parameter $\lambda = 0.16$). However, even if these results are promising, the system should only be used as a warning system for the teacher, who should always have the final say.

An interesting aspect to note about the combination strategy C_s, is that it takes time into account with $\lambda = 0.1$, weighing recent assignments more than older ones. Since $\tau(t)$ measures in months, this means that a recent assignment gets $e^{12 \cdot 0.1} \approx 3.3$ times the weight of a one year old assignment. This corresponds well with the idea that high school students writing style changes over time, as also observed in [6].

When looking at the low false accusation rates of Figure 2 (right), one have to consider two things before translating them into practice: a) T_{test} is balanced, while in reality much less than half of assignments are written by a ghostwriter, and b) ghostwriting does happen, also in our data set, and thus most likely some of our labels are wrong. A possible remedy for the second point could be to adjust FN to $\mathrm{FN} - \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\gamma\mathrm{T}$ (where γ is the estimated fraction of ghostwriters and $\mathrm{T} = \mathrm{TP}+\mathrm{FN}$), and similar for TP, under the assumption that a negative sample and a corrupted positive sample are indistinguishable. Adjusting for this would obviously lead to improved accuracy and false accusation rate, but requires a good estimate of γ.

4 Conclusion

We achieved an accuracy of 0.875, with a false accusation rate of 0.141 and a catch rate of 0.896. We show how false accusation rate can be improved at the cost of catch rate and accuracy. Results are good enough for practical use, and even with a slightly lower catch rate, the system is still expected to have a preventive effect. However, one has to keep in mind that, in practice, the data set is not 50/50 balanced, which obviously will affect the results. Making a split imitating the real world is hard for two reasons: one needs a good approximation of the actual fraction of ghostwriters, and even if this fraction is known, the number of corrupt labels would be approximately the same as the number of negatives, making it impossible to beat a false accusation rate of 0.5, even for a perfect classifier. Finding a clean data set or establishing ground truth would alleviate these problems, and could be interesting prospects for future work.

Another interesting direction is to analyze writing style changes over time more in depth, motivated by the chosen combination strategy and preliminary experiments, which show how two texts written within a shorter time span have higher similarity on average.

References

[1] Politisk flertal vil gøre salg af eksamensopgaver ulovligt. `http://nyheder.tv2.dk/politik/2017-06-21-politisk-flertal-vil-gore-salg-af-eksamensopgaver-ulovligt`. Accessed: 2018-11-25.

[2] Efstathios Stamatatos. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556, March 2009.

[3] Douglas Bagnall. Author Identification using multi-headed Recurrent Neural Networks. In *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*. CEUR-WS.org, September 2015.

[4] Alberto Bartoli, Alex Dagri, Andrea De Lorenzo, Eric Medvet, and Fabiano Tarlao. An author verification approach based on differential features. In *CEUR WORKSHOP PROCEEDINGS*, volume 1391. CEUR, 2015.

[5] Chen Qian, Tianchang He, and Rao Zhang. Deep learning based authorship identification. 2018. report, Stanford University.

[6] Niels Dalum Hansen, Christina Lioma, Birger Larsen, and Stephen Alstrup. Temporal context for authorship attribution: a study of Danish secondary schools. In *Multidisciplinary information retrieval*, pages 22–40. Springer, 2014.

[7] John Burrows. 'Delta': a Measure of Stylistic Difference and a Guide to Likely Authorship. *Literary and Linguistic Computing*, 17(3):267–287, 2002.