Style Change Detection
NLP Course Project

Matteo Donati, Ildebrando Simeoni and **Diego Biagini**
Master's Degree in Artificial Intelligence, University of Bologna
{matteo.donati10, ildebrando.simeoni, diego.biagini2}@studio.unibo.it

Abstract

This paper describes two simple but effective approaches for the three Style Change Detection (SCD) tasks for PAN at CLEF 2022. The first approach we propose is based on fine-tuning the BERT transformer and training different classification heads so to solve all three tasks in an end-to-end fashion. The second approach, instead, is based on using the Sentence-BERT (SBERT) transformer and cosine similarity so to solve the given classification problems by working directly in the embedding space implied by the chosen transformer-based encoder. We show how, using both approaches, we are able to achieve adequate results and also outperform existing state-of-the-art methods in a particular task.

1 Introduction

The SCD problem belongs to the author identification class of problems, and it is the only means to detect plagiarism in a document if no comparison texts are given. Likewise, style change detection can help to uncover gift authorships, to verify a claimed authorship, or to develop new technology for writing support.

The goal of the SCD-2022 tasks is to identify text positions within a given multi-author document at which the author switches (Zangerle et al., 2022). In particular, given a document combined from the StackExchange questions and answers, participants are asked to solve three tasks: 1). Style Change Basic: for a text written by two authors that contains a single style change only, find the position of this change (i.e., cut the text into the two authors' texts on the paragraph-level); 2). Style Change Advanced: for a text written by two or more authors, find all positions of writing style change (i.e., assign all paragraphs of the text uniquely to some author out of the number of authors assumed for the multi-author document); 3). Style Change Real-World: for a text written by two or more authors, find all positions of writing style change, where style changes now not only occur between paragraphs, but at the sentence level. All the possible scenarios are shown in figure 1

In all the previous editions of the competition, the joining teams tried to tackle the given problems by proposing a mixture of techniques, including ensembles of models, feature-based and similarity-based approaches (Zangerle et al., 2021, 2020, 2019). Based on the results of such approaches, it is possible to notice how end-to-end neural models are the models that best perform at solving the given tasks. This motivates us to explore the use of pre-trained transformer models to address the authorship analysis issue for writing style changes. Indeed, pre-trained models can be used to solve similar problems, using transfer learning techniques to fine-tune the pre-trained model to fit a particular downstream task.

In this paper we propose two similar but different approaches that try to solve all the three SCD-2022 tasks in a unified manner, by relying on neural models that incorporate pre-trained encoders such as BERT (Devlin et al., 2018) and SBERT (Reimers and Gurevych, 2019). In particular, the first approach takes into consideration the BERT encoder so to produce contextual embeddings that can be fed to fully-connected or convolutional layers and produce the final probability distribution that determines whether a couple of paragraphs or sentences of text is written by the same author. The second approach, instead, considers the SBERT encoder so to produce embeddings that can be directly compared using cosine similarity, and thus determining whether two paragraph or sentences of text are written by the same author.

The performances of the two approaches have been measured on the given validation set, using macro F_1-score. In particular, compared to the current state-of-the-art, both of the aforementioned approaches worked well, achieving adequate scores.

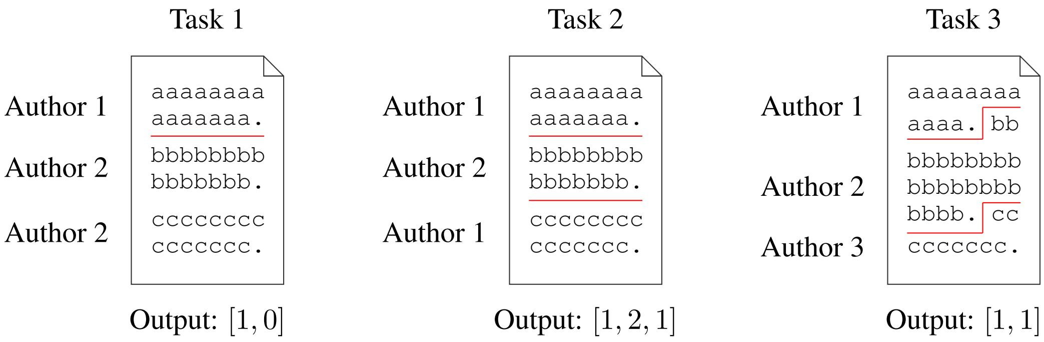


Figure 1: Possible scenarios and expected outputs

Moreover, the SBERT approach outperformed existing state-of-the-art methods at solving the Task 1 problem.

2 Background

The SCD-2018 edition focused on detecting whether a document is single-authored. This task can be treated as a binary classification problem. In this edition, different approaches were used to detect style changes, including ensembles of supervised learning models (Zlatkova et al., 2018) and feature-based approaches (Safin and Ogaltsov, 2018; Khan, 2018).

The SCD-2019 edition added a task for detecting the actual number of authors within a single document. One successful approach (Nath, 2019) was based on clustering algorithms, so to yield a number of clusters corresponding to the number of authors.

The SCD-2020 edition added another task based on detecting style changes between two consecutive paragraphs. In this case, a paragraph representation based on character, lexical and syntactic features was proposed, using a clustering algorithm for style change detection (Castro-Castro et al., 2020). Another group used the BERT model to generate embedding representations and used such embeddings to train a random forest classifier to detect style changes between two paragraphs (Iyer and Vosoughi, 2020).

The SCD-2021 edition also added a task to detect the number of authors and locating specific author changes within the text. Style change and author identification were regarded as binary classifications based on similarity measurements (Zhang et al., 2021).

Lastly, the SCD-2022 edition introduced the Style Change Real-World task, in which style changes can also occur at the sentence level. The work we present with this paper is deeply inspired by last editions' solutions. Indeed, both of our approaches use famous neural language models and use the computed embeddings to train specific classifiers or to compute similarity scores between texts.

The main innovation we tried to bring is the use of metric learning and siamese networks to compute similarity measurements in the vector space implied by SBERT. Indeed, the aim of metric learning is to compare low-dimensional embeddings of some input so to find semantically similar inputs in an effective way. One important type of neural models that is capable of finding similar inputs, as stated by the definition of metric learning, is the siamese neural network. This network is virtually composed of two backbones that share weights and work with two different input vectors to compute comparable output vectors. An example of such network is the SBERT model. SBERT is a modification of the BERT model that uses siamese and triplet networks to derive semantically meaningful embeddings that can be compared using similarity measures. The authors of SBERT describe this model as a single BERT model that has been trained using siamese as well as triplet architectures, depending on the structure of the data at hand.

After a careful analysis of the capabilities of such network, we made sure to properly manipulate the given datasets so to fine-tune SBERT using both the contrastive loss (Hadsell et al., 2006) as well as the triplet objective function described by Reimers and Gurevych, 2019. Indeed, the idea behind metric learning is to use specific losses to guide the model to favor a clustered structure of the embeddings such that: 1). the distance between semantically similar inputs is minimized; 2). the distance between semantically different inputs is maximized.

3 System description

Regarding the first approach, the pipeline we implemented is the following. We started by analysing all the three SCD-2022 datasets and by creating appropriate tables that could fit our needs. In particular, each of the computed tables is composed of $N \times 7$ elements, where N is the number of rows of the particular table and represents the number of documents in the original dataset. Each row of such table is composed of seven elements: `authors`, the total number of authors in the generic row (i.e. document); `site`, the source site from which the document was extracted; `multi-author`, a boolean flag indicating whether the generic document has been written by multiple authors; `changes`, a list of boolean varibles determining changes between paragraphs or sentences; `paragraph-authors`, a list of integer variables that label each paragraph or sentence with an author ID; `input_text`, the content of the entire document; `splitted_text`, the list of paragraphs or sentences of the generic document.

After having computed the aforementioned tables, we noticed how the given datasets, especially the one related to task 1, were not balanced. To solve this problem, we preprocessed the data so to be able to effectively train our models. The techniques we used to preprocessed the data are described, in more details, at section 4.

Then, we defined our BERT-based models. All the layers that define these models are shown in table 1. In particular, the BERT version we used is the BERT base cased version[1]. The layers of the classification head of each m_x^1 model were inspired by some successful works presented in the current edition of the SCD challenge (Lao et al., 2022; Lin et al., 2022). Indeed, the classification heads we defined are either composed of fully-connected layers or composed of uni-dimensional convolutional layers. Moreover, in order to be able to detect style changes between two different paragraphs or sentences, we had to manipulate the aforementioned dataframes so to incorporate two different consecutive paragraphs or sentences within the same BERT input. To do this, we used the official BERT tokenizer, that uses the special `[CLS]` and `[SEP]` tokens to build the different inputs. Thus, the inputs of the model are list of tokens of the form `[CLS] P/S-A [SEP] P/S-B [SEP]`, where `P/S-A` corresponds to the encoding of a paragraph/sentence, and `P/S-B` to the encoding of the subsequent paragraph/sentence. Other than the aforementioned list of token, for each possible pair of texts we produced: 1). a boolean mask, having the same length of the input tokens list, that determines at which positions the input list of tokens contains padding; 2). a boolean list, having the same length of the input tokens list, that determines the positions of `P/S-A` and of `P/S-B` within the input list.

The last bit of data manipulation was related to casting the Style Change Advanced problem (i.e. task 2), to be a binary classification problem, as in the case of task 1 and task 3. This was made possible by applying data augmentation to the validation set of task 2, thus considering all possible pairs of paragraphs. By doing this, we were able to tackle all the three tasks by using the same architectures and methodologies.

All the m_x^1 models have been trained for the same number of epochs, using a specific learning rate scheduler that properly modifies the learning rate within the same epoch. During the training procedure, the BERT layers of each encoder have been fine-tuned on the particular tasks at hand, so to produce contextual embeddings that are relevant for our specific problems. More information on the entire setup can be found in section 5.

Once trained, the models have been tested using the validation sets provided by the authors of the SCD-2022 challenge. In particular, the performances of each model were measured using the macro F_1-score. Lastly, considering the results on task 2 and task3 (namely, the two tasks that present a total number of authors within $[1, 5]$), some error analysis was made. In this last phase, we wanted to study the performances of each model with respect to the total number of authors of the given documents, and to compare the four m_x^1 models based on the trends we obtained from this study.

4 Data

5 Experimental setup and results

6 Discussion

7 Conclusion

[1] https://huggingface.co/bert-base-cased

Model	Encoder	Classifier
m_1^1	BERT	Global Max-Pool, FC-2
m_2^1	BERT	Global Avg-Pool, FC-2
m_3^1	BERT	Global Max-Pool, FC-128, FC-2
m_4^1	BERT	(Conv1D-128, Max-Pool) $\times$ 3, Flatten, FC-2

Table 1: BERT-based models

Model	$\cdots$
m_1^2	$\cdots$
m_2^2	$\cdots$
m_3^2	$\cdots$
m_4^2	$\cdots$

Table 2: SBERT-based models

8 Links to external resources

References

Daniel Castro-Castro, Carlos Alberto Rodríguez-Losada, , and Rafael Munoz. 2020. Mixed style feature representation and b_0-maximal clustering for style change detection. *PAN at CLEF 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Aarish Iyer and Soroush Vosoughi. 2020. Style change detection using bert. *PAN at CLEF 2020*.

Jamal Ahmad Khan. 2018. A model for style change detection at a glance notebook for pan at clef 2018. *PAN at CLEF 2018*.

Qidi Lao, Li Ma, Wenyin Yang, Zexian Yang, Dong Yuan, Zhenlin Tan, and Langzhang Liang. 2022. Style change detection based on bert and conv1d. *PAN at CLEF 2022*.

Tzu-Mi Lin, Chao-Yi Chen, Yu-Wen Tzeng, and Lung-Hao Lee. 2022. Ensemble pre-trained transformer models for writing style change detection. *PAN at CLEF 20122*.

Sukanya Nath. 2019. Style change detection by threshold based and window merge clustering methods. *PAN at CLEF 2019*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Kamil Safin and Aleksandr Ogaltsov. 2018. Detecting a change of style using text statistics. *PAN at CLEF 2018*.

Eva Zangerle, Maximilian Mayerl, Martin Potthast, and Benno Stein. 2020. Overview of the style change detection task at pan 2020. *PAN at CLEF 2020*.

Eva Zangerle, Maximilian Mayerl, Martin Potthast, and Benno Stein. 2021. Overview of the style change detection task at pan 2021. *PAN at CLEF 2021*.

Eva Zangerle, Maximilian Mayerl, Martin Potthast, and Benno Stein. 2022. Style change detection 2022.

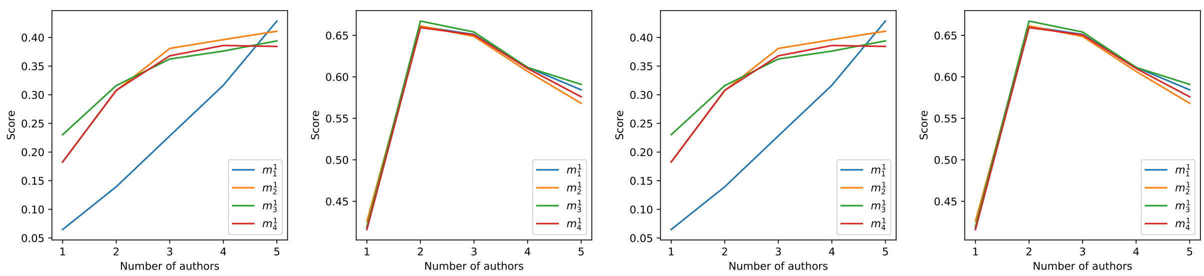
Eva Zangerle, Michael Tschuggnall, Gunther Specht, Benno Stein, and Martin Potthast. 2019. Overview of the style change detection task at pan 2019. *PAN at CLEF 2019*.

Zhijie Zhang, Zhongyuan Han, Leilei Kong, Xiaogang Miao, Zeyang Peng, Jieming Zeng, Haojie Cao, Jinxi Zhang, Ziwei Xiao, and Xuemei Peng. 2021. Style change detection based on writing style similarity. *PAN at CLEF 2021*.

Dimitrina Zlatkova, Daniel Kopev, Kristiyan Mitov, and Atanas Atana. 2018. An ensemble-rich multi-aspect approach for robust style change detection. *PAN at CLEF 2018*.

Model	Task 1	Task 2	Task 3
m_1^1	0.72	0.28	0.66
m_2^1	0.71	0.45	0.66
m_3^1	0.71	0.44	0.66
m_4^1	0.68	0.43	0.66
m_1^2	0.70	0.33	0.58
m_2^2	0.78	0.44	0.67
m_3^2	0.78	0.42	0.66
m_4^2	$\mathbf{0.80}$	0.41	0.64
Lin et al., 2022	0.74	$\mathbf{0.54}$	$\mathbf{0.73}$

Table 3: Macro F_1-scores



(a) m_x^1 on task 2
(b) m_x^1 on task 3
(c) m_x^2 on task 2
(d) m_x^2 on task 3

Figure 2: Macro F_1-scores with respect to number of authors