Style Change Detection on Real-World Data using an LSTM-powered Attribution Algorithm

Notebook for PAN at CLEF 2021

Robert Deibel[1], Denise Löfflad[1]

[1]Eberhard Karls University Tübingen, Geschwister-Scholl-Platz, 72074 Tübingen, Germany

Abstract

The task of Style Change Detection (SCD) aims at detecting author switches within one document based on their individual writing style. In this notebook, this task is divided into three sub-tasks: detecting multi-authored documents, finding style change positions, and attributing each paragraph to a unique author [1, 2]. We chose different machine learning approaches for the first task of multi-author detection, and the second task of style change detection. Our approach to the third task of SCD-based authorship attribution is a hybrid method building upon the prediction of the style change detection and extended by an attribution algorithm. The data was given by the PAN'21 challenge and is a data set collected from an English written Q&A forum [1, 2]. While the approach to task three showed to be very computationally expensive, we found good results for task one and two with F1-scores of 86% for task one and 78% for task two on the validation set.

Keywords

Style Change Detection, Stylometry, Word Embeddings, NLP, Machine Learning, LSTM, MLP

1. Introduction

Due to the ever increasing interconnection and collaborations resulting in multi-authored texts, identifying authors within texts becomes an interesting effort. The steadily increasing amount of readily available data gathered form online text forums or short message boards allows for construction of methods and models for text analysis based on machine learning approaches. Possible applications for such models can relate to e.g. plagiarism detection or style analysis. Style analysis opens the possibility for authors to adjust their writing styles to that of other collaborators and consequently to write more coherent texts.

The task of Style Change Detection (SCD) aims at analysing texts by detecting whether or not a document is multi-authored, and if so, where a Style Change occurs in order to find author changes and predict authorship in the context of one document [3]. This year's PAN Shared Task builds up on competitions from previous years [3]. The PAN'21 SCD task [1, 2] is threefold:

1. Determining whether a document is single or multi-authored
2. Determining whether a Style Change occurs between paragraphs
3. Assigning each paragraph to one author

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

deibel.robert@gmail.com (R. Deibel); denise.loefflad@student.uni-tuebingen.de (D. Löfflad)

Especially the last task, which is a type of author attribution task, adds additional complexity in comparison to the previous years. Traditionally in author attribution, the setting contains a fixed and known set of authors. The data set for this year's PAN Shared Task represents real-world data [1, 2] and therefore, we decided to treat the exact number of authors of the texts as unknown. Additionally there is no closed set of authors thus we have to assume that the sets of authors for any two texts are pairwise disjoint. Therefore, we can only use a single text to identify the authors, and cannot use a global author profile, that is one spanning all texts.

The following notebook focuses on our efforts to construct machine learning based methods to solve the three given tasks. After this introduction we will give a brief overview of related works in Section 2, describe our methods in Section 3, state our results and their evaluation in Section 4, and lastly give a short discussion and conclusion in Section 5.

2. Related Works

As mentioned before, the goal of the PAN'21 Shared Task is to detect the exact positions of authorship changes. In order to lead up to this task, it is split into three sub-tasks: deciding whether a document is multi-author, finding style changes between paragraphs, and lastly attributing the paragraphs uniquely to one of the authors.

Regarding the first task of detecting multi-authored documents, several different solutions have been proposed over the last decades [4, 5, 6]. Zuo *et al.* [6] used a multi-layer perceptron (MLP) with a single layer for the binary classification task and represent each document by a TFIDF-weighted word vector. Iyer and Vosoughi [5] generated sentence vectors using embeddings and compared Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, and Naive Bayes classifiers. They reported to yield the best results with a Random Forest classifier.

Similar to the first task, the second task received attention by researchers over the last years. Especially in the PAN Shared Task series, the detection of style changes between paragraphs has been addressed repeatedly and various approaches have been proposed. Iyer and Vosoughi (2020) [5] yielded the best results in 2020 [3], using Google's BERT language model and word embeddings [7]. Another approach included computing nearly 200 textual features and using B_0-maximal clustering [8], however this approach yielded less promising results in the PAN Task. Moreover, Hosseinia *et al.* (2018) [4] implemented a complex Parallel Hierarchical Attention Network using LSTM layers and achieved promising results.

Finally, the third task, namely assigning paragraphs to specific and unknown authors, is a more recent problem and has consequently been addressed less frequently. It is similar to the problem of *authorship identification of multi-author documents* (AIMD) [9]. Sarwar *et al.* [9] tackled this problem by using character n-grams and a multilingual feature space to identify co-authors from a set of known authors.

In all three tasks, individual writing style plays a role and can be addressed more or less specifically. Castro-Castro *et al.* (2020) [8] for example computed nearly 200 textual features for their pipeline. Stylometry is defined as *statistical analysis of written texts* [10] and assumes that textual features are quantifiable and represent one's *individual and distinct* writing style reliably. Stylometry has application in and has been shown to work effectively for Authorship Attribution [11, 12, 13, 10], and for authorship attribution of single author documents [9].

In the next section, we will focus on our contribution to the research in Style Change Detection and Author Attribution, while taking into account past research and results.

3. Methods

For the three tasks tackled in this notebook we chose different machine learning approaches for the first task of multi-author detection, and the second task of style change detection. We attempted to solve the first task using per-document embeddings and an MLP, while we utilized per-paragraph embeggins and textual features as input for an LSTM for the second task. Our approach for the third task of SCD-based authorship attribution is a hybrid method building upon the prediction of the style change detection and extended by an attribution algorithm. The training of the models was performed on an Intel®Core™i7-7500U CPU at 2.70GHz without hardware acceleration. Additionally, a server environment on TIRA [14] was provided by the PAN'21 team.

In the following we will describe our approaches in more detail and state other important considerations for our application.

3.1. Data Set

The data set provided by the PAN'21 SCD challenge consisted of one training and one validation set with 11200 and 2400 problems, respectively, as well as solutions for these problems. The problems consist of uniquely English posts scraped from the StackExchange network that where put together in one to four author documents of total length between 1000 and 3000 characters. Additionally to the training and validation sets a test set was generated by the PAN'21 team but not provided during development. The test set is similar to the validation in terms of size and content [1, 2]. Overall the data set is similar to the one used in Zangerle *et al.* [3]. Since the data sets provided additional metadata e.g. the total number of authors in a text, it would have been possible to construct a model that uses this additional data for its prediction. To simulate the real world application of SCD we chose not to consider the maximum number of authors in our application but rather keep this value variable.

3.2. Textual Features

We manually extracted a small set of traditional features that are often used in Authorship Attribution tasks [15], and one additional feature. We computed these measures for every paragraph. The measures are described in more detail as follows:

- *Corrected Type-Token Ratio (CTTR)*: The total number of unique words (types) divided by the total number of words (tokens) in a paragraph. In contrast to the traditional TTR which is negatively influenced by texts longer than 100 words, the CTTR takes into account the varying lengths of the paragraphs and is therefore not affected by text length.

- *Mean Sentence Length in Words*: The average number of words in a sentence. Considering the creation of the data set - an online Q&A, where presumably each author is less likely to

stick to a writing standard - we expected this measure to be adequate for this dataset and task.

- *Mean Word Length*: The average word length in syllables. It has been shown that the use of shorter compared to longer words is a good indicator of an author's style [16].

- *Function Word Frequency*: It is unlikely that the frequency of function word use is consciously controlled [17]. We decided to add this feature, as it is not expected that their frequency would vary much with the topic of the text.

- *Linsear Write Formula*: A readability formula to score the difficulty of English text. The standard Linsear Write metric runs on a 100-word sample. For calculating it, we used the following method: For the first 100 words of the text, for each simple word, defined as words with two syllables or less, we added one point to the result r. For each complex word, defined as words with three syllables or more, we added three points to r. Then, we divide the points by the number of sentences in the 100-word sample, and adjust the result r:

$$LWF = \begin{cases} \frac{r}{2}, & \text{if } r < 20 \\ \frac{r}{2-1}, & \text{otherwise} \end{cases}$$

We chose to manually implement those rather basic features as they have been shown to be good predictors of style. However, we did not expect the model trained on this set of five features to outperform the model trained on word embeddings, simply because of the seemingly small amount of information that such a set would achieve.

3.3. Embeddings

The word embeddings used for our model are the pretrained fastText word vectors provided by Facebook's AI Research lab [18]. They were trained on Common Crawl and Wikipedia data, using continuous bag of words with position-weights, in dimension 300, with character n-grams of length five, a window of size five and ten negatives. Unlike Word2Vec, fastText uses character n-grams in order to create an inherent association between words that share the same stem, thus it not only encodes semantic and syntactic information, but morphological information as well.

For the task of multi-author prediction we chose to calculate the embeddings on a per-document basis. We anticipated that multi-author documents would be separate from single-author documents in embedding space.

For the task of style-change detection we use a per-paragraph embedding. Since each paragraph is guaranteed to have been written by a single author, we computed the average of the word embeddings for each paragraph, thus generating a paragraph embedding. As pointed out by Kenter *et al.* [19], simply averaging word embeddings of all words in a text has proven to be a surprisingly successful and efficient way of obtaining features across a multitude of tasks. The final vectors of both tasks were then padded and fed into the model.

3.4. Model Single vs. Multiple

Due to the promissing results reported by Hosseinia *et al.* (2018) [4], we decided to opt for a MLP to tackle the first task of the challenge. Our machine learning models across all the tasks are built using the Keras API [20] with the TensorFlow [21] backend. For this approach specifically we used a standard MLP pipeline with three hidden, fully connected, and feed forward layers. We assumed that the set of multi-authored documents and the set of single authored documents could be well separated in the space of per-document embeddings. As a per-document approach makes more sense for this task, we decided not to use textual features. Computing textual features on a per-document basis would distort the feature values and lead to poorer results.

The number of neurons and the learning rate were determined using a grid search approach in the range of 32 to 512 with increments of 32 for the number of neurons and the options of 10^{-2}, 10^{-3} and 10^{-4} for the learning rate. For the hidden layers this resulted in 480, 480, 288 neurons, respectively. The optimal learning rate was found to be 10^{-3}.

The activation of hidden layers was set as the ReLU function as this has proven to be successful in MLP applications [22]. The output layer activation was set as the sigmoid function as we want to predict a binary decision. Again for all of our models the Adam optimizer [23] and the binary cross entropy loss was used as the optimization loss function.

As input data we used only the per-document embeddings as described in Section 3.3 as we wanted to analyze the discriminative abilities of MLPs on per-document level. We believe that calculating per-document complexity measures would not yield a satisfying separation since the documents are constructed to be similar in their global structure.

3.5. Model Style Change Basic

We trained a two-layered Bidirectional LSTM model with 128 hidden units per layer, adding a Masking layer, and a Time-Distributed layer as the output layer with a sigmoid activation function. We used binary cross-entropy as the loss function. In order to prevent overfitting, early stopping was used.A threshold of 0.5 was established in order to decide whether there is a style change. We applied normalization. We found that a batch size of one achieved the best results, compared to batch sizes five and ten, which yielded results worse by 60% points.

Our anticipation for using an LSTM was that the model could learn similarities in writing style on a per-paragraph basis by using the paragraphs as time steps in its input.

3.6. Model Style Change Real-World

We implemented different approaches for this task, which turned out to be more difficult than the previous two tasks. Our first approach was to train a simple LSTM model similar to that of task two, but because there are no consistent classes across the data, the model achieved bad results. The second approach was a two-fold pipeline containing a k-means clustering algorithm and a classification model. As the data for the Shared Task should represent real-world data the number of authors of every text is unknown and may differ from text to text. Therefore, it was not straightforward to build a clustering system that would predict the number of authors.

The third approach showed to yield the best result. For this approach, we used an LSTM-powered Attribution Algorithm, visualized in Figure 1 and represented as pseudo-code in Al-

gorithm 1. The algorithm functions as follows. For every paragraph the authorship attribution decision is made. We utilize the style change detection prediction that is generated by our LSTM-model for the whole document. The first paragraph is always attributed to author A_1, we assume that the paragraphs are written by the same author, A_1, as long as no style change occurs. After a style change was detected in the predictions we construct a new prediction problem for our LSTM to solve. This problem consists of the current paragraph up for authorship attribution and preceded by a previously attributed paragraph p_i. p_i is iterated until either, *no* style change is detected or p_i is equal to the current paragraph. The first case implies that the author of p_i is the same in the current paragraph while the second case implies that the author was not detected before and a new author should be selected.

Algorithm 1 LSTM-powered Attribution Algorithm

$T := setOfTexts()$
$A \leftarrow \{newAuthor()\}$
for all $t \in T$ **do**
    $P := setOfParagraphs(t)$
    $p_1 := firstPragraph(P)$
    $setAuthor(p_1, A_1)$
    $P' \leftarrow \{p_1\}$
    for all $p \in P$ **do**
        if No Style Change **then**
            $setAuthor(p, A_{p-1})$
        else
            for all $p' \in P'$ **do**
                if Style Change **then**
                    continue
                else
                    $setAuthor(p, A_{p'})$
                    break
                end if
            end for
            if Author Not Set **then**
                $setAuthor(p, newAuthor())$
            end if
        end if
        $P' \leftarrow P' \cup \{p\}$
    end for
end for

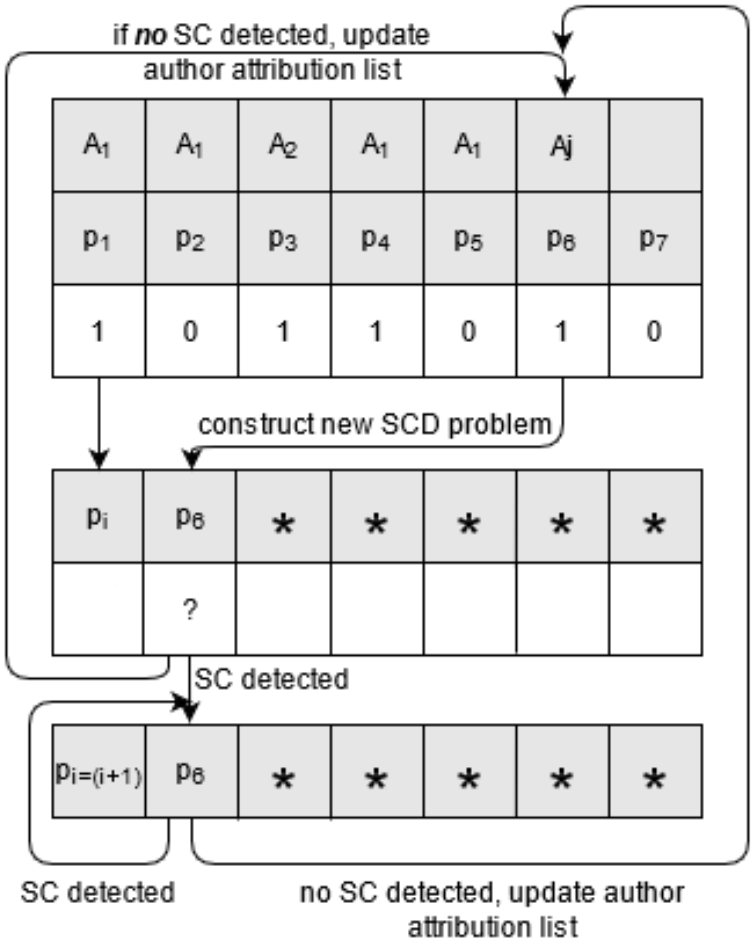

Figure 1: Visualization of one iteration of the attribution of our proposed algorithm. A_j represents the attributed author. p_i represents i-th paragraph. The white box represents the prediction of the trained model, the $*$ represent padding. First the style change prediction of the whole document is analyzed. Every paragraph that is classified as *no style change (0)* is considered to have the same author as before. For a detected style the previous paragraphs are iterated and a style change problem is constructed with the iterated paragraph p_i as the first entry and the paragraph that the author is to be detected. *no* style change leads us to assume that we have found the author. A style change leads us to repeat the process until we find an author or the algorithm ends up at the current paragraph. In the last case a new author is chosen.

4. Evaluation

In this section, we present the results we achieved by testing the models on the validation set that was made available by the PAN'21 committee [1, 2], followed by the results yielded on the test set. For the test set, only the F1-scores are reported by the PAN team.

Table 1
F1, Precision, Recall, Accuracy Scores for our Predictions for Task 1 on Validation and Test Set using Embeddings

	Validation Set	Test Set
F1	86.86	62.08
Precision	91.88	-
Recall	82.37	-
Accuracy	79.16	-

Table 2
F1, Precision, Recall, Accuracy Scores for our Predictions for Task 2 on Validation and Test Set

	Feature Model	Embeddings Model	Combined Model	Combined Model Test Set
F1	74.61	78.00	79.18	66.90
Precision	85.34	82.51	87.26	-
Recall	66.27	74.07	72.47	-
Accuracy	94.87	95.91	95.95	-

4.1. Single vs. Multiple

In this section we present our results for the task of detecting multi-authored documents, shown in Table 1. It can be seen that the model yields good results across all measures. The small drop in recall and accuracy suggest that the model classifies some multi-authored documents as single-authored documents. When evaluated on the test set, the F1-score drops noticeably, this can indicate that the model might not generalize well.

Nevertheless, the F1-score achieved by our model outperforms the winning's team model from last year's PAN Task [3, 5] and therefore suggests that this method is a good approach to the task of detecting multi-authored documents. Of course, the data set differs somewhat from that of Iyer and Vosoughi (2020) [3, 5], but we still believe that the results are comparable. The generation of the data was similar, and since both data sets are retrieved from the same English Q&A forum, the data sets are also comparable.

4.2. Style Change Detection Basic

In this section we present our results for the task of detecting style change positions. As can be seen in Table 2 the three categories yielded similar results. It is interesting to see that the model trained with the textual features only achieved high accuracy. Nevertheless, the comparatively low recall score of 66.27% suggests that this model tend to predict no style change, the reason for the high accuracy being the imbalance in the data. The other two models trained with the embeddings and the combination of the embeddings and features achieved recall scores of 74% and 72.5%, respectively. This suggests that these models are more robust to imbalance in the data and therefore more suitable for real-world data.

Compared to the self-reported results from the winning team of last year's PAN Shared Task,

our model does not outperform Iyer and Vosoughi's model [5]. Similar to Task 1, a drop in the F1-score evaluated on the test is observed.

Nevertheless, all three models yield good results and despite the drop in recall, the feature model achieves satisfying results.

4.3. Style Change Real-World

The implementation of the algorithm showed to be very computationally expensive and time consuming. Due to our limited access to computational power we were not able to run this code on the validation set in order to get results. Nevertheless, when running it on a smaller corpus we yielded promising outcomes. However, considering the small amount of data we used we decided to refrain from reviewing the results here.

The PAN team reported a F1-score of 26.25 %. Considering this result, we assume that a sequence of two paragraphs might not be enough information for the model to make a legitimate prediction.

5. Conclusion and Outlook

In this notebook, we presented our approaches to solve the PAN'21 Shared Task of Style Change Detection [1, 2]. For each sub-task, we implemented different algorithms and built a hybrid algorithm for the task of assigning paragraphs to authors. This task showed to be the most complex out of the three.

With regards to the stylometric aspect of the problems, we implemented only five different and rather traditional features. Surprisingly, it was possible to achieve good results with only those features, which shows that for this data set, the individual style of the authors was accurately representable by mean word length, mean sentence length, function word frequency, CTTR, and the LWF.

Our approaches to the first and second task yielded good results on the validation set, which shows that simple machine learning models can be good solutions for the tasks. Future work could try to implement attention based models [24], convolutional layers combining and compressing paragraphs, or an autoencoder approach [25]. Expecially autoencoders are often used in style transformation [26, 27] and could be a potential candidate for style change tasks.

The third task is a newer problem and more complicated to solve, especially on real-world data with an unknown number of unknown authors. For future projects, we plan on attempting to implement a clustering/classification model for this problem. Furthermore, it is possible to build up on our model by increasing the number of paragraphs used to make predictions and parallelizing the loop of the attribution algorithm to increase computation speed. We therefore consider our model to be a promising approach despite achieving low results. It is also important to note that we challenged ourselves to a more real-world-like problem by not taking into account the maximum number of authors, as it would certainly lower the complexity of the task if this information is used.

References

[1] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection, in: 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[2] E. Zangerle, M. Mayerl, , M. Potthast, B. Stein, Overview of the Style Change Detection Task at PAN 2021, in: CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[3] E. Zangerle, M. Mayerl, G. Specht, M. Potthast, B. Stein, Overview of the style change detection task at pan 2020, CLEF, 2020.

[4] M. Hosseinia, A. Mukherjee, A parallel hierarchical attention network for style change detection: Notebook for pan at clef 2018., in: CLEF (Working Notes), 2018.

[5] A. Iyer, S. Vosoughi, Style change detection using BERT, in: CLEF, 2020.

[6] C. Zuo, Y. Zhao, R. Banerjee, Style change detection with feed-forward neural networks., in: CLEF (Working Notes), 2019.

[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[8] D. Castro-Castro, C. A. Rodríguez-Losada, R. Muñoz, Mixed style feature representation and b0-maximal clustering for style change detection (2020).

[9] R. Sarwar, N. Urailertprasert, N. Vannaboot, C. Yu, T. Rakthanmanon, E. Chuangsuwanich, S. Nutanong, *cag*: Stylometric authorship attribution of multi-author documents using a co-authorship graph, IEEE Access 8 (2020) 18374–18393.

[10] H. Ramnial, S. Panchoo, S. Pudaruth, Authorship attribution using stylometry and machine learning techniques, in: Intelligent Systems Technologies and Applications, Springer, 2016, pp. 113–125.

[11] M. Bhargava, P. Mehndiratta, K. Asawa, Stylometric analysis for authorship attribution on Twitter, in: International Conference on Big Data Analytics, Springer, 2013, pp. 37–47.

[12] D. I. Holmes, Authorship attribution, Computers and the Humanities 28 (1994) 87–106.

[13] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, D. Woodard, Surveying stylometry techniques and applications, ACM Computing Surveys (CSUR) 50 (2017) 1–36.

[14] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.

[15] P. Muller, Style change detection, ETH Zurich (2019).

[16] M. Koppel, J. Schler, S. Argamon, Computational methods in authorship attribution, Journal of the American Society for information Science and Technology 60 (2009) 9–26.

[17] C. Chung, J. W. Pennebaker, The psychological functions of function words, Social communication 1 (2007) 343–359.

[18] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157 languages, arXiv preprint arXiv:1802.06893 (2018).

[19] T. Kenter, A. Borisov, M. De Rijke, Siamese cbow: Optimizing word embeddings for

sentence representations, arXiv preprint arXiv:1606.04640 (2016).
[20] F. Chollet, et al., Keras, https://keras.io, 2015.
[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: https://www.tensorflow.org/, software available from tensorflow.org.
[22] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org.
[23] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.
[24] A. M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, 2015. arXiv:1509.00685.
[25] M. J. Kusner, B. Paige, J. M. Hernández-Lobato, Grammar variational autoencoder, in: International Conference on Machine Learning, PMLR, 2017, pp. 1945–1954.
[26] Y.-J. Zhang, S. Pan, L. He, Z.-H. Ling, Learning latent representations for style control and transfer in end-to-end speech synthesis, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 6945–6949. doi:10.1109/ICASSP.2019.8683623.
[27] D. Ramani, S. Karmakar, A. Panda, A. Ahmed, P. Tangri, Autoencoder based architecture for fast & real time audio style transfer, 2018. arXiv:1812.07159.