Rich Style Embedding for Intrinsic Plagiarism Detection

Oumaima Hourrane[1], El Habib Benlahmer[2]
Laboratory Information Technology and Modeling,
Faculty of Science Ben Msik,
Hassan II University of Casablanca, Morocco

***Abstract*—Stylometry plays an important role in the intrinsic plagiarism detection, where the goal is to identify potential plagiarism by analyzing a document involving undeclared changes in writing style. The purpose of this paper is to study the interaction between syntactic structures, attention mechanism, and contextualized word embeddings, as well as their effectiveness on plagiarism detection. Accordingly, we propose a new style embedding that combines syntactic trees and the pre-trained Multi-Task Deep Neural Network (MT-DNN). Additionally, we use attention mechanisms to sum the embeddings, thereby experimenting with both a Bidirectional Long Short-Term Memory (BiLSTM) and a Convolutional Neural Network (CNN) max-pooling for sentences encoding. Our model is evaluated on two sub-task; style change detection and style breach detection, and compared with two baseline detectors based on classic stylometric features.**

Keywords*—*Plagiarism detection; style embedding; deep neural network; stylometry; syntactic trees

I. Introduction

The enormous availability of textual data and documents via the Internet (web pages, databases, digital documents, etc.) makes it more straightforward for unauthorized copying and stealing of intellectual properties of others. To counter this phenomenon, there are two main approaches to detect plagiarism that have been studied and implemented; extrinsic and intrinsic plagiarism detection [3]. The former method consists of comparing suspicious passages or documents against a range of external sources, while the latter method focuses on using local analysis to detect suspicious passages within a document by comparing their writing styles without using any referenced document.

The act of collecting reference set of documents for the extrinsic plagiarism detection is very troublesome from the computational time point of view. In addition, a source of a plagiarized document may not always be available. Therefore, researchers in recent works had addressed these limits by the intrinsic plagiarism detection, which has been introduced first by Meyer Zu Eissen and Stei [10]. The idea is that one would expect that a plagiarized passage from a document is stylistically inconsistent from the non-plagiarize passages, without comparing a suspicious document to the potential sources. Such stylistic inconsistencies can be detected by quantifying certain stylometric features, such as character n-gram profile [11], Parse trees [12], word n-gram and word frequency [13].

Since the major advances in Deep learning-based pre-trained word embeddings [14], [15], [16], It became fundamental to many Natural Language Processing tasks to use these representations as input features to other models [5], [6]. Furthermore, the emergence of the contextualized language representations [17], [20], [19], [18], [4] along with Multi-task Learning concept [21], had given birth to more refinded language representations, the Multi-Task Deep Neural Network (MT-DNN) [1]. By training this latter on various tasks such as Natural Language Inference, Parsing task, and Neural Machine Translation, several semantic and syntactic properties have been learned.

In this paper, our goal is to leverage the properties learned from the pre-trained MT-DNN, and to make our style representation richer by adding along with the MT-DNN another type of embeddings based on syntactic trees, where each word is represented by POS tags from the leaf to the root for the constituency tree, and the dependencies links for the dependency tree. Then, the resulted sequence are served as input to a Bi-LSTM architecture to get the final syntactic embeddings of words. To combine the strengths of the three type of embeddings we employ a weighted sum based on a self-attention mechanism [22]. Finally, we train a fully connected layer followed by a softmax on top of the sequence encoding to classify each sequence. The latter can be a document for the style change detection sub-task, and a sentence for the style breach detection sub-task, more details can be found in Section III.

The rest of this paper is organized as follows: In Section 2, we describe our architecture with more details. In Section 3 we outline our experimental settings and compare our result with two baseline methods. In Section 4, we introduce some related works on intrinsic plagiarism detection. Finally, we summarize our work and future directions.

II. Related Work

In this section, we report previous works that have been done for the intrinsic plagiarism detection task, and other related tasks, such as Style Breach detection [9], and Style Change Detection [7].

By considering the intrinsic plagiarism detection task as a one-class classification problem, [24] construct a wide range of stylometric features and apply a density method to separate outliers from target class, where they assumed that the outliers to be the plagiarized passages from the source documents. They as well employed the meta-learning approach of Koppel and Schler [25] to post-process unreliable stylometric analysis results.

To compute writing style differences, [26] proposed an intrinsic algorithm that describes a new variant by using stop words with TF-weighting as input feature, after the document segmentation into equal size chunks and features representation, they compare each chunk representation with that of the whole suspicious document.

To to find style change and irregularities, [12] analyzed syntactic information of authors focusing on sentence constructions. Namely, they extract POS-sequences representations, next they construct a distance matrix of every distinct pair sentences by applying sequence alignment algorithm. Finally, they employed Gaussian normal distribution function over the mean distances, all thresholds and parameters are optimized by implementing genetic algorithms.

The work of [27] consists of running various step for the intrinsic plagiarism detection. Beginning by chunking to suspicious document into equal size overlapping windows, then they represent each chunk by the relative frequencies of a predetermined set of high-frequency character trigrams. Next, they measure the distance between consecutive windows using is a symmetric adaptation of the normalized distance. Finally, they employed an algorithm for outliers detection based on Principal Components Analysis.

There are two approaches that deal with the style breach detection instead of detecting suspicious passages to be plagiarized, The first approach [28] assume that each sentence vector depends on the previous and next sentence vectors, so they proposed an architecture based on mapping sentences by using a pre-trained encoder-decoder as an embedding. The resulting vectors are used then for the outliers detection step. The second approach is an unsupervised analysis [8], it is focused on retrieving various stylistic features along with some new features naming common English word frequencies.

These following works propose models that answer the question of if a given document had a style change within a document; [30] use character-based Convolutional Neural Networks (CNN) which can be applied to any language. [31] combine TF.IDF representation of the documents with other stylistic features, they then use an ensemble of diverse models including SVM, Random Forest, AdaBoost, MLP and LightGBM. [32] employ two parallel attention networks, feeding the model with the hierarchical structure of the language using a pre-trained statistical parser.

Finally, [33] propose a feature discretization method based on two-step cluster for Naive Bayes. Namely, they used the TF-IDF and query language model as a discrete feature and False Positive/False Negative (FP/FN) threshold to improve the accuracy.

III. Methodology

Our work consist of using different kind of embeddings for the intrinsic plagiarism detection problem. We apply our architecture on two sub-tasks. The first, is the style change detection, where we verify if an input document contains style changes, therefore, it contains sections written by other authors. The second is the style breach detection, where we detect the intrusive sections that are stylistically deviant from the main writing style.

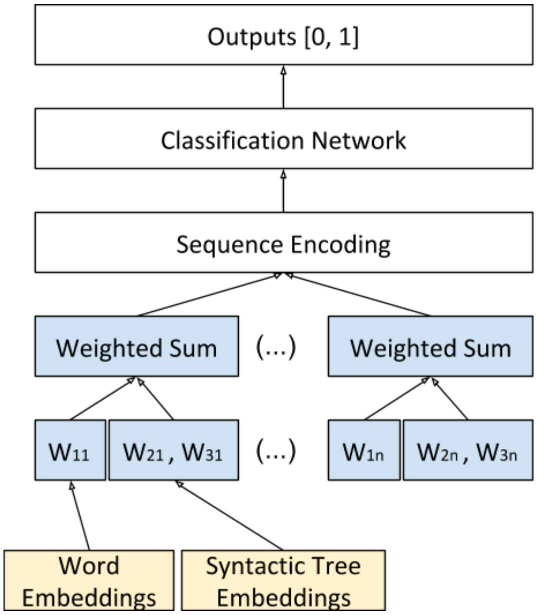

Fig. 1. General Architecture for Intrinsic Plagiarism Detection.

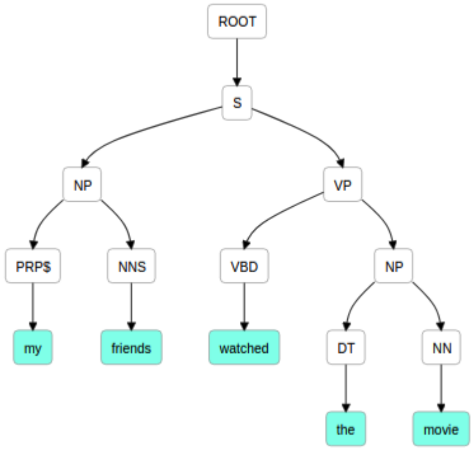

Fig. 2. An example of the constituency tree of 'My friends watched the movie'.

The general framework of our model is illustrated in Fig. 1. Here the input of the model is the embedding of a given sequence while the output indicate respectively if a sequence is plagiarized or not. We describe in the next sections more details about each elements of our architecture.

A. Style Embeddings (SE)

1) Constituency Trees Embedding (CTE): We extract constituency trees for each sentence is a document using the Stanford CoreNLP [29]. We choose this kind of syntactic tree to describe the grammatical aspect of a given text, as it defines the way to hierarchically construct a sentence from words based on constituency relations. Fig. 2 shows an example of a constituency tree.

For each word in a sentence, We define constituency tree feature as a path starting from the root to the corresponding

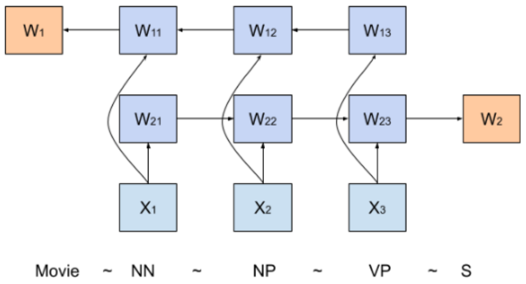

Fig. 3. Constituency tree embedding for a given word using a Bi-directional LSTM.

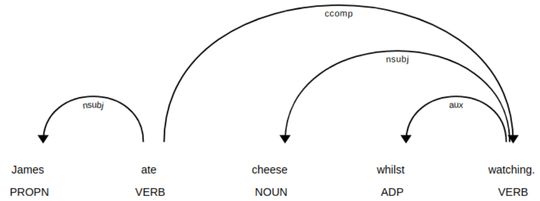

Fig. 4. An example of the dependency tree of 'James ate cheese whilst watching.'.

leaf, for example, the word 'movie' in our example, can be represented by a syntactic sequence (NN, NP, VP).

As it has been proven effective in [34], we construct syntactic embedding by using a Bi-directional LSTM to represent a constituency tree feature of a given word as indicated in Fig. 3. the input is index vectors for syntactic tags initialized with a multivariate normal distribution. Finally, each word is represented with the concatenation of W_1 and W_2; the output vectors of the forward and backward LSTMs, respectively.

2) Dependency Trees Embedding (DTE): The dependency tree is a syntactic tree constructed by dependency grammars, namely, in a given sentence, words are connected to each other by direct links called dependencies.

The advantage of this structure is its ability to deal with language that is morphologically rich and has a free word order, which represents only the information that is necessary for the sentence. Fig. 4 illustrates an example of a dependency tree.

Relations among the words is represented above the sentence with directed and tagged arcs from root to leaves. For each word we encode its dependency sub-tree with its dependent nodes ordered by their positions in the original sentence, using Bi-directional LSTM as indicated in Fig. 5. Similar to the constituency tree, each word is represented with the concatenation of W_1 and W_2; the output vectors of the forward and backward LSTMs respectively.

3) Contextualized word embedding: The second input that we fed into our model is the embedding information of words we denote it an embedding of a single word as W_3. We choose to use the pre-trained MT-DNN [1], which has created new state-of-the-art results across many popular NLP benchmarks. The particularity of this model is that is a one-to-many multi-tasking learning framework that computes the loss across different tasks and applies them at the same time. Moreover,

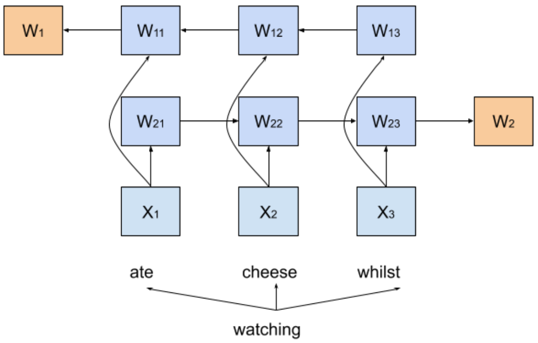

Fig. 5. Dependency tree embedding for a given word using a Bi-directional LSTM.

several stylistic properties have been learned in this model by training it on a parsing task, Natural Language Inference and Neural Machine Translating tasks.

4) Sequence Encoding: For a sequence of n words $\{p_j\}_{j=1}^{n}$, we denote the three embeddings we get from the previous step as $\{W_{i,j}\}_{j=1}^{n}$ and $i = 1, 2, 3$. As our aim is to apply the attention weights within the sequence, we then combine these embedding by taking their weighted sum.

$$W_j' = \sum_{i=1}^{3} \alpha_{i,j} W_{i,j}$$

Where $\alpha_{i,j}$ are scalar weights from a self-attention mechanism [22]: $\alpha_{i,j} = \phi(a.W_{i,j} + b)$, where a and b are learned parameters and ϕ is a Softmax function.

Next, we experiment with two encoders CNN and a standard bidirectional LSTM (BiLSTM-Max) with max-pooling to obtain a global sequence embedding.

For the CNN encoder, a convolution operation involves the application of a filter k to a window of h words to produce a new feature. For instance, $c_{j,i}$ is the i^{th} feature generated from a window of the word embedding $W'_{j:j+h-1}$

$$c_{j,i} = f(a_i.W'_{j:j+h-1} + b_i)$$

Where a_i and b_i respectively are the weight and bias f the j^{th} filter. Finally, a max-pooling operation is applied over these features to get the final sequence representation E_{CNN}^{seq}:

$$E_{CNN}^{seq} = max_{row}(c_{j,i})$$

Secondly, the BiLSTM-Max computes for each direction two sets of m hidden states as follows:

$$\overrightarrow{h_i} = \overrightarrow{LSTM_i}(W_1, W_2, ..., W_i)$$

$$\overleftarrow{h_i} = \overleftarrow{LSTM_i}(W_i, W_{i+1}, ..., W_m)$$

In order to get the final hidden states, for each time-step the hidden states are afterward concatenated, then a max-pooling operation is applied over their components to get the final sequence representation: $h = max([\overrightarrow{h_i}, \overleftarrow{h_i}]_{j=1,2,..,m})$

B. Style Change Detection

This task deals with a particular question, whether or not a document has multiple authors. Certainly, the changes of authorship have to be determined by capturing changes in writing styles, and the problem can be taken by applying a binary classification over the whole document. In that case, each document is represented as a fixed-sized vector of the resulted style embedding, this vector is next fed into a fully connected layer followed by a Softmax layer to predict the existence of style changes. we kept the same dimension of the fully connected layer as the sequence embedding from the previous step. The outputs of the network are 0 or 1. A binary cross-entropy loss function is used to train the entire model. This later is optimized using Adam optimizer [23].

C. Style Breach Detection

This task consists of finding sections within a document where the authorship changes. This problem can simply be seen as a text segmentation problem based on the writing style. In our work, we assume that style breaches may occur on sentences ending so we apply our style-embedding on the sentence level. Then, we applied a sentence outliers detection as commonly used in Intrinsic Plagiarism Analysis [24], where we determine if an input sentence is plagiarized or not. same as in the style change detection sub-task, we fed the resulted style-embedding of the input sentences into a fully connected layer followed by a Softmax layer, then we train and optimize the model by using a binary cross-entropy loss function and Adam optimizer, respectively.

IV. Experiment

A. Datasets

We evaluate the performance of the proposed approach for the style breach detection sub-task on PAN-PC-11 corpus[1], which is addressed for the intrinsic Plagiarism detection. It contains artificially plagiarised passages for each suspicious document and mentions the offsets of plagiarised and non-plagiarized parts. The next sections describe more details about the experiment.

The second corpus we used in the style change detection sub-task is PAN 2018 corpus[2] based on user posts from QA network Stack Exchange. It contains 2980 training problems, 1492 validation problems and 1352 test problems, where for each subset the amount of documents combining styles changes is equal to the number of documents containing no changes.

B. Baseline Models

Concerning the style change detection sub-task, we chose as a baseline model, the method of Zlatkova [2] that is state-of-the-art on PAN 2018 Competition [7]. The authors used stacking ensemble architecture. After the text pre-processing, they chunked the text into three equal fragments and used several lexical and syntactical features, and trained four different classifiers on each fragment, they combined these models with TF-IDF based gradient boosting model and fed them into a Logistic regression meta-classifier to produce the final result.

For the style breach detection sub-task, we chose as a baseline model the approach of Khan [8] which is state-of-the-art on PAN 2017 Competition [9]. The authors used as stylometric feature most frequent POS tags and words, and other dictionaries. Next, they compute the similarity score between each two adjacent sliding windows which is then compared to a predefined threshold in order to decide the style breach between two sentences if it exists.

C. Experimental Settings

Syntactic information related to constituency trees and dependency trees were extracted from Stanford CoreNLP[3]. For a given syntactic tree representation of a single word we chose a window size of max 10 tags in the constituency tree vector and 10-nearest dependents in the dependency tree vector, in order to prevent excessive usage of the memory, while benefiting the performance of the model [22].

As described in Section III-A4, we use a pre-trained contextualize word embedding MT-DNN [1] to serve as part of the input into the model. Moreover, all the embedding types we used in the input have a dimension of 1000.

For the classification network, as indicated in Section III-B. We train a fully connected layer followed by a Softmax on top of the sequence embedding after the max pooling operation. A binary cross-entropy is used as loss function. The initial learning rate is set to 10^-4, dropout to 0.2 and we use Adam for optimization.

D. Results

The overall performance results are depicted in Table I. Our purpose is to apply the combination of different embeddings, though, we report the accuracy of each embeddings to see which one contributes in the final performance, we also report the difference between using CNN max pooling operation for sequence encoding and BiLSTM-Max architecture.

Concerning the style change detection sub-task. We could see that the SE-Combined-BiLSTM achieved the best result with an accuracy of nearly 88%, which has shown that using BiLSTML max pooling operation for sequence encoding achieve good performance for this problem. The same thing on the style breach detection task, the SE-Combined-BiLSTM outperform the other models with an accuracy of 80% Table II, which suggest that using both BiLSTM for sequence embedding is most effective to extract the plagiarized passages that are stylistically different from the original text.

In both sub-tasks, we could see that The SE-DTE Dependency trees embedding outperform the other representations in term of accuracy, followed by SE-CTE Constituency tree embedding, then the SE-MT-DNN Contextualized word embedding. And when combining the three embedding types, the model achieved higher accuracy, providing the evidence that combining a contextualize word embedding with syntactic information can accurately detects the style change with documents. Therefore detects plagiarism intrinsically. Moreover, we

[1] https://webis.de/data/pan-pc-11.html

[2] https://pan.webis.de/clef18/pan18-web/author-identification.html#style-change-detection

[3] https://stanfordnlp.github.io/CoreNLP/

TABLE I. PERFORMANCES OF ALL MODELS ON THE STYLE CHANGE DETECTION SUB-TASK

Models	Accuracy
SE-CTE	0.70
SE-DTE	0.76
SE-MT-DNN	0.69
SE-Combined-CNN	**0.78**
SE-Combined-BiLSTM	0.88
[2]	0.75

TABLE II. PERFORMANCES OF ALL MODELS ON THE STYLE BREACH DETECTION SUB-TASK

Models	Accuracy
SE-CTE	0.66
SE-DTE	0.73
SE-MT-DNN	0.70
SE-Combined-BiLSTM	**0.80**
SE-Combined-CNN	0.76
[8]	0.59

found that BiLSTM-Max for the sequence encoding operation has better performance than their CNN counterparts in both sub tasks, which suggests that BiLSTM can more effectively preserve the syntactic and stylometric information.

V. CONCLUSION AND FUTURE WORKS

In this work, we proposed an architecture for the intrinsic plagiarism detection, based on a rich style representation detailed in Section III. We applied our approach on two sub-task; Style change detection and style breach detection, and evaluate it on PAN-18 and PAN-PC-11 corpus, and compare it with baseline detectors. The performance of the combined style embedding is promising, providing the evidence that combining a contextualize word embedding with syntactic information can accurately detects plagiarism intrinsically within a document. Roughly speaking, We subsequently show how our style embedding approach can be used to shed new light on the usage of structural embedding along with other state-of-the-art word embeddings in stylometry.

There are many future directions to improve our intrinsic plagiarism detector including but not limited to: In future work, it would be interesting to apply our idea to different tasks, such as Authorship Attribution and Extrinsic Plagiarism Detection, in order to explore what kinds of embeddings are most useful for each plagiarism detection tasks. It would also be interesting to further exploit more stylometric features, such as word frequencies and word/character n-grams in our combined style representation, and to apply them on an unsupervised plagiarism detection task that focus on separating the original sequences from the plagiarized ones. In addition, it would be interesting to examine the effect of adding the attention weight within and after the sequence encoding, the difference between the embedding concatenation and the weighted variant of our method, and the attention weights change during the training.

REFERENCES

[1] Liu X, He P, Chen W, Gao J. *Multi-task deep neural networks for natural language understanding.* arXiv preprint arXiv:1901.11504. 2019 Jan 31.

[2] Zlatkova D, Kopev D, Mitov K, Atanasov A, Hardalov M, Koychev I, Nakov P. *An ensemble-rich multi-aspect approach for robust style change detection.* CLEF 2018 Working Nots of CLEF. 2018.

[3] Hourrane, Oumaima, and El Habib Benlahmar. "Survey of plagiarism detection approaches and big data techniques related to plagiarism candidate retrieval." Proceedings of the 2nd international Conference on Big Data, Cloud and Applications. ACM, 2017.

[4] Hourrane, Oumaima, et al. "Using Deep Learning Word Embeddings for Citations Similarity in Academic Papers." International Conference on Big Data, Cloud and Applications. Springer, Cham, 2018.

[5] Mifrah, Sara, and El Habib Ben Lahmar. "Semantico-automatic Evaluation of Scientific Papers: State of the Art." Proceedings of the 2nd international Conference on Big Data, Cloud and Applications. ACM, 2017.

[6] Omar Zahour, El Habib Benlahmar, Ahmed Eddaoui and Oumaima Hourrane, "Automatic Classification of Academic and Vocational Guidance Questions using Multiclass Neural Network" International Journal of Advanced Computer Science and Applications(IJACSA), 10(10), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101072

[7] Kestemont, Mike, Michael Tschuggnall, Efstathios Stamatatos, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. *Overview of the author identification task at PAN-2018: cross-domain authorship attribution and style change detection.* In Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al., pp. 1-25. 2018.

[8] Khan, Jamal Ahmad. *Style Breach Detection: An Unsupervised Detection Model.* In CLEF (Working Notes). 2017.

[9] Tschuggnall, Michael, Efstthios Stamatatos, Ben Verhoeven, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. *Overview of the author identification task at PAN-2017: style breach detection and author clustering.* In Working Notes Papers of the CLEF 2017 Evaluation Labs/Cappellato, Linda [edit.]; et al., pp. 1-22. 2017.

[10] Zu Eissen, Sven Meyer, and Benno Stein. *Intrinsic plagiarism detection.* European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2006.

[11] Stamatatos E. *A survey of modern authorship attribution methods.* Journal of the American Society for information Science and Technology. 2009 Mar;60(3):538-56.

[12] Tschuggnall, Michael, and Günther Specht. *Plag-inn: Intrinsic plagiarism detection using grammar trees.* International Conference on Application of Natural Language to Information Systems. Springer, Berlin, Heidelberg, 2012.

[13] Oberreuter G, VeláSquez JD. *Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style.* Expert Systems with Applications. 2013 Jul 1;40(9):3756-63.

[14] Mikolov T, Chen K, Corrado G, Dean J. *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781. 2013 Jan 16.

[15] Pennington, Jeffrey, Richard Socher, and Christopher Manning. *Glove: Global vectors for word representation.* Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[16] Joulin A, Grave E, Bojanowski P, Mikolov T. *Bag of tricks for efficient text classification.* arXiv preprint arXiv:1607.01759. 2016 Jul 6.

[17] Kiros, Ryan, et al. *Skip-thought vectors.* Advances in neural information processing systems. 2015.

[18] Devlin, Jacob, et al. *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805 (2018).

[19] Howard, Jeremy, and Sebastian Ruder. *Universal language model finetuning for text classification.* arXiv preprint arXiv:1801.06146 (2018).

[20] Peters, Matthew E., et al. *Deep contextualized word representations.* arXiv preprint arXiv:1802.05365 (2018).

[21] Zhang, Yu, and Qiang Yang. *A survey on multi-task learning.* arXiv preprint arXiv:1707.08114 (2017).

[22] Lin, Zhouhan, et al. *A structured self-attentive sentence embedding.* arXiv preprint arXiv:1703.03130 (2017).

[23] Kingma, Diederik P., and Jimmy Ba. *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980 (2014).

[24] Stein, Benno, Nedim Lipka, and Peter Prettenhofer. *Intrinsic plagiarism analysis.* Language Resources and Evaluation 45.1 (2011): 63-82.

[25] Koppel, Moshe, and Jonathan Schler. *Authorship verification as a one-class classification problem.* Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.

[26] Oberreuter G, L'Huillier G, Ríos SA, Velásquez JD. *Approaches for intrinsic and external plagiarism detection.* Proceedings of the PAN. 2011.

[27] Kestemont M, Luyckx K, Daelemans W. *Intrinsic plagiarism detection using character trigram distance scores.* Proceedings of the PAN. 2011.

[28] Safin, Kamil, and Rita Kuznetsova. *Style Breach Detection with Neural Sentence Embeddings.* CLEF (Working Notes). 2017.

[29] Manning, Christopher, et al. *The Stanford CoreNLP natural language processing toolkit.* Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. 2014.

[30] Schaetti, Nils. *Character-based Convolutional Neural Network for Style Change Detection.* Training 2980.1490: 1490.

[31] Zlatkova D, Kopev D, Mitov K, Atanasov A, Hardalov M, Koychev I, Nakov P. *An ensemble-rich multi-aspect approach for robust style change detection.* CLEF 2018 Working Nots of CLEF. 2018.

[32] Hosseinia, Marjan, and Arjun Mukherjee. *A Parallel Hierarchical Attention Network for Style Change Detection.*

[33] Wijaya, Adi, and Romi Satria Wahono. *Two-Step Cluster based Feature Discretization of Naïve Bayes for Outlier Detection in Intrinsic Plagiarism Detection.* Journal of Intelligent Systems 1.1 (2015): 1-8.

[34] Liu, Rui, et al. *Structural embedding of syntactic trees for machine comprehension.* arXiv preprint arXiv:1703.00572 (2017).