Experiments with Convolutional Neural Networks for Multi-Label Authorship Attribution

Dainis Boumber[*], **Yifan Zhang**[*], **Arjun Mukherjee**
Department of Computer Science, University of Houston
Philip Guthrie Hoffman Hall, 3551 Cullen Blvd., Room 501, Houston, TX 77204-3010, USA
{dboumber, yzhang114}@uh.edu, arjun@cs.uh.edu

Abstract

We explore the use of Convolutional Neural Networks (CNNs) for multi-label Authorship Attribution (AA) problems and propose a CNN specifically designed for such tasks. By averaging the author probability distributions at sentence level for the longer documents and treating smaller documents as sentences, our multi-label design adapts to single-label datasets and various document sizes, retaining the capabilities of a traditional CNN. As a part of this work, we also create and make available to the public a multi-label Authorship Attribution dataset (MLPA-400), consisting of 400 scientific publications by 20 authors from the field of Machine Learning. Proposed Multi-label CNN is evaluated against a large number of algorithms on MLPA-400 and PAN-2012, a traditional single-label AA benchmark dataset. Experimental results demonstrate that our method outperforms several state-of-the-art models on the proposed task.

Keywords: multi-label authorship attribution, convolutional neural networks, datasets

1. Introduction

Authorship Attribution uses textual features to distinguish between texts written by different authors (Stamatatos, 2009). A typical AA problem is a single-label text-categorization task: given a set of candidate authors for whom text samples of undisputed authorship are available, a text is assigned to one candidate author (Sebastiani and Ricerche, 2002; Stamatatos, 2009). Less common yet equally important is the case that involves identifying multiple co-authors of a document. This is the problem of multi-label AA and as we will see it is significantly harder than classical AA. We briefly review classical AA and then introduce our work in multi-label AA.

Classification algorithms utilizing lexical, semantic, syntactic, stylistic, and character n-gram features have been explored by Graham et al. (2005), Gamon (2004), Sapkota et al. (2015), and Shrestha et al. (2017). Qian et al. (2014) proposed a tri-training method to solve AA under limited training data per author. It extended standard co-training using three views: lexical, character and syntactic and was shown to have better generalization performance. This method assumes that a large set of unlabeled documents authored by the same given closed set of authors is available. Sapkota et al. (2016) leveraged Domain Adaptation in an AA scenario where articles on different topics may be written by the same author and labeled training data is limited. The method introduced was a modification of Structural Correspondence Learning (Blitzer et al., 2006) and requires a large set of unlabeled documents pertaining to the target domain and written by the same authors. Seroussi et al. (2012) used latent topic features to improve attribution. Although useful, it requires a large text collection per author. AA via text distortion has been used on traditional PAN corpora (Stamatatos, 2017). AA with a large number of authors and limited training data has been studied by Luyckx and Daelemans (2008). A lazy memory-based learner based on k-NN was shown to work well with combinations of features. Ruder et al. (2016) used character level and hybrid multi-channel Deep Neural Networks for large scale multi-author AA to a great degree of success; however, their work was concerned with classical single-label scenarios only.

We propose a CNN for multi-label AA tasks. Our design treats a document as a set of sentences where each one has multiple labels. Individual authorship of continuous sections is taken into account together with the possibility of co-authors influencing each other's style or editing passages written by others. We name this strategy *collaborative section attribution*. Multi-label CNN utilizes depth-wise convolutions for the separate processing of the two input channels, capturing information unique to each, which helps filters activate on more relevant inputs. We conduct a series of experiments using our model, the recently successful version of CNN by Kim (2014) and a large number of baselines. Proposed Multi-label CNN outperforms the competition by a significant margin on multi-label data (MLPA-400) and matches or defeats relevant baselines on single-label tasks (PAN 2012).

For evaluation, we consider a realistic problem of multi-label AA in the realm of scientific publications by creating a publicly available dataset consisting of 400 Machine Learning papers, Machine Learning Papers' Authorship 400 (MLPA-400)[1]. To the best of our knowledge, multi-label AA of scientific publications has not received a lot of attention. It deserves more attention because automatic resolution of authorship issues in papers can have a variety of downstream applications in intellectual property managements, citation analysis, archival systems, and author disambiguation. The task is challenging: papers have many authors whose writing style can evolve or influenced

[*]The first two authors contributed equally to the work but are named alphabetically.

[1]https://github.com/dainis-boumber/AA_CNN/wiki/MLPA-400-Dataset

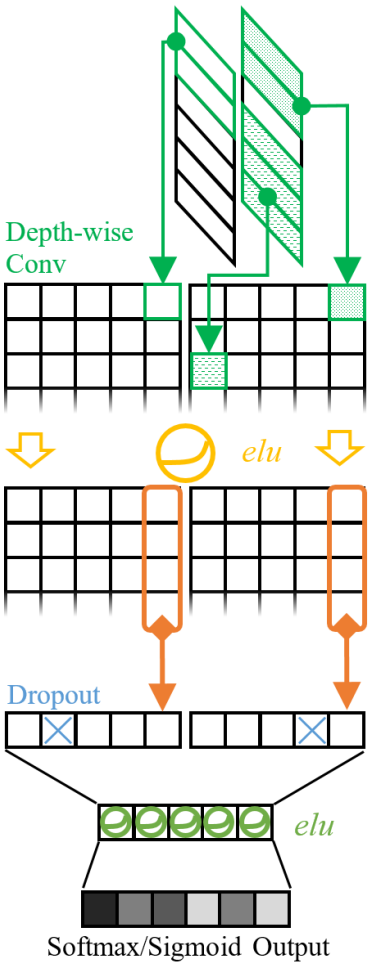

Figure 1: The architecture of proposed multi-label CNN.

by colleagues, they contain direct quotes from other works, authors' contribution to the paper in terms of the amount of text written is unknown; the number of papers and authors is large.

The contribution of our work is threefold: a CNN that employs collaborative section attribution and separate channels in depth-wise convolutions, a novel real-world multi-label MLPA-400 corpus from top cited ML authors for an AA scenario, a thorough performance evaluation of the proposed algorithm with relevant baselines on the new MLPA-400 dataset and PAN-2012.

2. Network Hierarchy

At a high level, our design is a multilayer CNN that either computes a probability distribution for an entire document (single-label problems) or an average of probability distributions over individual sentences of a document (multi-label problems). Attributing multi-author work sentence by sentence intuitively makes sense, because co-authors typically write different sections of the paper.

2.1. CNN Architecture

Let the document to be classified be $\boldsymbol{D}$. Each $\boldsymbol{D}$ consists of $|S|$ number of sentences and each sentence consists of $|T|$ words. We classify each $\boldsymbol{S}$ and take the mean to obtain predicted label of $\boldsymbol{D}$. To allow the network to generate consistent dimensions, we pad sentences to the same length $|T| = 128$. Documents are padded to same number of sentences $|S| = 128$. The ground truth label associated with each document is denoted using vector $\boldsymbol{y}$. Then $|\boldsymbol{y}|$ is equal to the cardinality of the set of possible authors. An element of $\boldsymbol{y}$ is marked 1 if the corresponding person is one of the authors and 0 otherwise.

To get an edge over Kim (2014), we use fixed word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) embeddings, leveraging Glove's superior general performance (Pennington et al., 2014) and word2vec's advantage when it comes to rare words or symbols (Shazeer et al., 2016). Words in $\boldsymbol{D}$ are mapped into pre-trained word2vec and Glove embedding space by replacing each word with a corresponding row in the embedding matrix $\boldsymbol{X}_{word2vec}$ and $\boldsymbol{X}_{glove}$, where both have $\boldsymbol{X} \in \mathbb{R}^{|V| \times n}$, assuming a total vocabulary count of $|V|$ and embedding dimension n. Hence replacing all word features in a sentence with one of the embedding matrices will transform the sentence into a matrix $\boldsymbol{S} \in \mathbb{R}^{|T| \times n}$.

To capture information that is distinctive to each embedding space, we help filters activate on relevant inputs by using depth-wise two-dimensional convolutions which process input from word2vec and Glove channels separately. In spirit with Kim (2014), multiple filters of multiple sizes h are used to extract features from $\boldsymbol{S}$. Unlike Kim, we use 100 filters for each window size $h \in \{1, 2, 3, 4, 5\}$ to let smaller filters pick up on simple stylistic features present, such as words unique to one author. Applying a filter $\boldsymbol{f} \in \mathbb{R}^{h \times n}$ on one of the input channels at word window i to $(i + h - 1)$ amounts to

$$c_i^f = elu\left(\boldsymbol{f} \cdot \boldsymbol{S}_{[i:i+h-1]} + b^f\right) \tag{1}$$

where b^f is the bias term corresponding to filter f and $elu(x)$ is the *exponential linear unit* (ELU). ELUs have negative values which pushes the mean of the activations closer to zero, resulting in faster training and lower variance. The positive part of these functions is the identity; their derivative is one and not contractive, thus the vanishing gradient problem is alleviated (Clevert et al., 2015).

We denote the number of filter sizes as $|\boldsymbol{h}|$ and the number of filters of each size as q. For $|\boldsymbol{h}| = 5$ and $q = 100$, we would have a total of 500 filters generating 500 features at each word location. By concatenating feature values generated by each filter into a vector, a total of 500 vectors $\boldsymbol{C}^f$s are generated for each sentence in each embedding channel.

$$\boldsymbol{C}^f = \left[c_1^f, c_2^f, \ldots c_{|T|-h+1}^f\right] \tag{2}$$

All vectors are batch-normalized (BN) to reduce overfitting (Ioffe and Szegedy, 2015).

Each $\boldsymbol{C}^f$ generated by the convolutional layer is max-pooled to keep only the largest value out of all the values across a sentence generated by one filter:

$$\hat{c}^f = max\left\{\boldsymbol{C}^f\right\} \tag{3}$$

All values resulting from max-pooling in both embedding channels are concatenated into a vector $\boldsymbol{o} \in \mathbb{R}^{|\boldsymbol{h}| \cdot q \cdot 2}$. Dropout (Srivastava et al., 2014) at the rate of 0.5 is then applied to $\boldsymbol{o}$ resulting in a vector $\hat{\boldsymbol{o}}$.

2.2. Predictions

Very short documents are unlikely to have more than one author. If the average document in the training corpus is shorter than the maximum sentence length $|T| = 128$, $\hat{\boldsymbol{o}}$ is passed to a softmax output layer for single-label classification. Otherwise, we pass $\hat{\boldsymbol{o}}$ to a fully connected sigmoid output layer of size $|\boldsymbol{y}|$, with one element per candidate author. For single-label tasks, *softmax cross entropy loss* was used. For multi-label tasks the loss used is *sigmoid cross-entropy*. We average resulting values for each sentence to determine the final document level result. The code can be obtained at `https://github.com/dainis-boumber/AA_CNN`

3. Datasets

3.1. MLPA-400

3.1.1. Considerations

Many approaches to creating a suitable corpus exist. For example, papers can be chosen across domains. However, even within one domain the stylistic differences between venues are significant enough to make individual style hard to detect. A random sample of authors can be taken, but the number of multi-labeled documents would be few. Another possibility is taking the transitive closure of the set of co-authors and extracting at least k papers per author. However, creation of such a dataset for any reasonable k results in a very large transitive set.

3.1.2. Design

Using Google Scholar as a source, we created a list of top 20 authors in Machine Learning, ranked by the number of citations. We ensured a reasonable number of papers had an overlap of authors (i.e., we also included papers that were jointly authored by the set of authors). For each author, 20 papers were downloaded for a total of 400 publications for the entire dataset. Each work is assigned 20 binary labels. The labels indicate which of the authors contributed to the paper's creation. 100 papers out of 400 have more than one author from the 20 listed. The number of authors ranged from 1 to 3 and the average was 1.2925. The text was extracted from the PDF files using pdfminer (Hinyama, 2017) and pre-processed. The title, authorship information, and bibliography fields were removed from each paper to ensure the classifier abides by the rules of blind review instead of simply using author list while learning authorship. Formulas, table and figure captions were retained as they may contain valuable author specific style and topic information. The dataset is available at `https://github.com/dainis-boumber/AA_CNN/wiki/MLPA-400-Dataset`

3.2. PAN-2012

For classical AA, we use the PAN-2012 (Juola, 2012) corpus and report performances on its 3 tasks: A, C and I. Their training sets consist of 2 documents per author. The test sets have 1 text per author, except Task A which has 2 texts from 3 authors, 800 to 6060 words each. Task C has 8 authors; the texts are larger, up to 13000 words long. Task I has 14 authors, with documents ranging from approximately 40,000 to 170,000 words. Further details on this data is available in (Juola, 2012).

4. Experiments

4.1. Baselines

Our method was tested against a wide array of baselines. We used n-grams with n ranging from 1 to 5 words and 1 to 8 characters. We experimented with TFIDF, hashing, count vectorization, binary bow model and doc2vec (Le and Mikolov, 2014). The resulting document vectors were used as inputs to the baseline classifiers: Support Vector Machine (SVM) (Cortes and Vapnik, 1995), Gaussian and Multinomial Naive Bayes (GNB and NB), Decision Tree with AdaBoost, Gradient Boosting, Random Forest, K Nearest Neighbors, Multilayer Perceptron (MLP) (Rumelhart et al., 1986), and Logistic Regression. We varied the hyper-parameters of the algorithms in order to achieve the best result (we vary kernels, the margin and the penalty between L_1, L_2, and $L_1 + L_2$ for hinge loss SVM with SGD; breadth and number of layers for MLP, etc.). The baselines were implemented using the scikit-learn library (Pedregosa et al., 2011).

We implemented CNN-non-static, a sentence classification approach proposed by Kim (2014) that has recently been successful in text classification. It initializes embedding layer with pre-trained word2vec vectors (Mikolov et al., 2013) and optimizes the embedding layer together with the rest of network's parameters during training. The CNN consists of one convolutional layer that samples from the input using multiple window sizes, a max over time pooling layer, and a fully connected output layer. Dropout (Srivastava et al., 2014) and L_2 are used for regularization.

The MLPA-400 problem is a multi-label task and cannot be directly solved by most classifiers. We used One vs. Rest Classification approach, fitting one binary classifier per class. We then associated a set of positive examples for a given class and a set of negative examples which represent all the other classes present within the training folds. No class balancing was performed to retain the natural class distribution in the data. As One vs Rest Classification scheme requires a separate model for each author, it is not feasible to use it with CNN-non-static on MLPA-400. Instead, we augmented CNN-non-static with the multi-label modification described in section 2.2. We evaluated an almost exhaustive combination of models with a total of 16 classifiers and 4 vectorizers that employ n-grams, with the maximum n equal to 5 for words and 8 for characters. This resulted in 16x4x5 + 16x4x8 or 832 baselines. We tested the CNN non-static and used doc2vec embeddings for each classifier, giving 33 more for a total of 865. After varying the hyper-parameters for those algorithms that allowed it, the total number of models was 1685.

For PAN-2012, the top performing teams results' in the AA challenge are also considered beyond the 1685 baseline models we trained.

4.2. Metrics

For the MLPA-400 data we use accuracy, micro and macro F1, Jaccard index and Hamming loss. In PAN-2012 exper-

Baseline	F1 macro	F1 micro	Jaccard index	Hamming loss	Accuracy
multi-label CNN (word2vec+glove)	0.736	0.744	**0.713**	0.031	**65.3%**
CNN-non-static (Kim, 2014)	0.685	0.695	0.664	0.037	60.8%
K Nearest Neighbors (binary 3-gram)	0.666	0.737	0.633	0.033	52.0%
Perceptron (sgd, l^2, binary 3-gram)	**0.748**	**0.751**	0.591	**0.030**	51.0%
SVM (sgd, squared hinge loss, l^2, binary 1-gram)	0.681	0.690	0.516	0.034	45.3%
MLP (doc2vec)	0.569	0.640	0.461	0.041	40.0%

Table 1: ML Papers evaluation results.

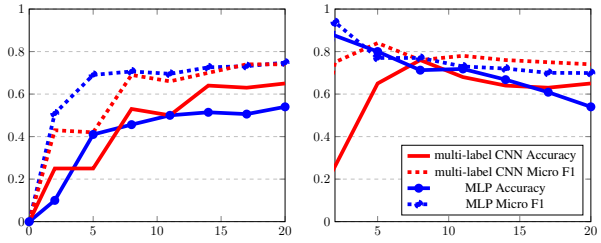

Figure 2: Varying the number of papers (left) and authors (right) on accuracy and micro-F1.

iments, we follow the instructions provided by the creators of the challenge and focus only on accuracy.

Exact label accuracy for multi-label problems is a particularly unforgiving metric: incorrectly attributing a single author from what can be a long list of co-authors results in the paper being marked as incorrectly classified. For this reason we also report Hamming loss and Jaccard index (also known as similarity coefficient), which are common metrics for multi-label tasks. Hamming loss is the fraction of labels that are incorrectly predicted. Jaccard index is the size of the intersection of the ground truths and the predictions divided by the size of their union.

5. Results and Discussion

5.1. MLPA-400

On this dataset, our approach is significantly more accurate, as seen in Table 1 yielding a confidence of $p = 0.0828$ against the next top competitor CNN non-static using a paired t-test across five-fold cross validation. Next, binary vectorization dominates all simple character and word vectorization approaches (not shown in Table 1 as they fall below top-500). Character n-grams perform poorly.

5.1.1. Effect of the number of Documents/Author

To discover the correlation between the amount of training data and the performance of our multi-label CNN, we explore the effect of the number of papers per author on accuracy and micro F1. The latter was chosen because it takes into account class and label balance. The # of papers per author was set to 2, then increased to 5, 8, 11, 14, 17 and 20. We compare multi-label CNN and MLP in Figure 2 (left). As the amount of training data (# of papers per author) available increases, so does the algorithms' ability to generalize. The improvement becomes minimal past 11 papers, but continues to increase. Because overfitting is reduced with the increased amount of training data (Brain and Webb, 1999), increasing the size of the dataset can benefit CNNs and other powerful models, allowing for deeper architectures that may discover new style features.

5.1.2. Effect of the number of Authors

To determine the effect of the number of authors on prediction accuracy, the number of papers was fixed at 20 per authors and the number of authors was varied between 2, 5, 8, 11, 14, 17 and 20. In Figure 2 (right), performance grows with the amount of training data, then decreases as the problem gets difficult due to the number of labels. Similar situations were also observed in Luyckx and Daelemans (2008) who recommend enriching the feature space by using combinations of features and employing a lazy learner.

5.2. PAN-2012

Task A: Our method, most baselines, and most competitors such as Sapkota and Solorio (2012) tied at 100% accuracy[2].

Task C: Our approach results in 100% classification accuracy. Only three of the challenge participants attain the same level of success (Grozea and Popescu, 2012; Sapkota and Solorio, 2012; Giraud and Artières, 2012). CNN-non-static performs very poorly, correctly classifying only 3 documents. SVM and NB achieve 87.5%, with other baselines falling far behind.

Task I: Kim-non-static and our method tie the state-of-the-art (Grozea and Popescu, 2012; Sapkota and Solorio, 2012; Tanguy et al., 2012) with the accuracy of 92.86%. SVM and NB score 85.71% and are on par with most of the contestants. Remaining baselines fall short.

5.3. Sensitivity of CNN Parameters

We found that using L_2 regularization improved training speed at the expense of accuracy. Additional convolutional layers failed to produce any effect.

6. Conclusion

This paper presented a CNN architecture designed to address multi-label Authorship Attribution problems. To test our design in non-traditional AA environment and alleviate the lack of relevant corpora, we created and made available to the public MLPA-400 — a dataset consisting of publications from well-known researchers. Experimental results show our method significantly outperforming the competition in a multi-label scenario and matching or surpassing state-of-the-art on traditional AA tasks.

Acknowldgments This work is supported in part by NSF 1527364.

[2]PAN-2012 results: http://pan.webis.de/clef12/pan12-web/pan12-authorship-attribution-evaluation-results.xlsx

7. Bibliographical References

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 120–128, Stroudsburg, PA, USA. Association for Computational Linguistics.

Brain, D. and Webb, G. I. (1999). On the effect of data set size on bias and variance in classification learning. In D. Richards, et al., editors, *Proceedings of the Fourth Australian Knowledge Acquisition Workshop (AKAW '99)*, pages 117–128, Sydney. The University of New South Wales.

Clevert, D., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *CoRR*, abs/1511.07289.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

Gamon, M. (2004). Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *COLING*.

Giraud, F.-M. and Artières, T. (2012). . In Pamela Forner, et al., editors, *CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy*, September.

Graham, N., Hirst, G., and Marthi, B. (2005). Segmenting documents by stylistic character. *Natural Language Engineering*, 11(4):397–415.

Grozea, C. and Popescu, M. (2012). Encoplot - Tuned for High Recall (also proposing a new plagiarism detection score). In Pamela Forner, et al., editors, *CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy*, September.

Hinyama, Y. (2017). Pdfminer: Python pdf parser and analyzer. `http://www.unixuser.org/~euske/python/pdfminer/`.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Juola, P. (2012). An overview of the traditional authorship attribution subtask. In Pamela Forner, et al., editors, *CLEF (Online Working Notes/Labs/Workshop)*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Luyckx, K. and Daelemans, W. (2008). Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 513–520, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Qian, T., Liu, B., Chen, L., and Peng, Z. (2014). Tri-training for authorship attribution with limited training data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 345–351, Baltimore, Maryland, June. Association for Computational Linguistics.

Ruder, S., Ghaffari, P., and Breslin, J. G. (2016). Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *CoRR*, abs/1609.06686.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA.

Sapkota, U. and Solorio, T. (2012). Sub-Profiling by Linguistic Dimensions to Solve the Authorship Attribution Task—Notebook for PAN at CLEF 2012. In Pamela Forner, et al., editors, *CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy*, September.

Sapkota, U., Bethard, S., y Gomez, M. M., and Solorio, T. (2015). Not all character n-grams are created equal: A study in authorship attribution. In Rada Mihalcea, et al., editors, *HLT-NAACL*, pages 93–102. The Association for Computational Linguistics.

Sapkota, U., Solorio, T., Gomez, M. M., and Bethard, S. (2016). Domain adaptation for authorship attribution: Improved structural correspondence learning. In *ACL (1)*. The Association for Computer Linguistics.

Sebastiani, F. and Ricerche, C. N. D. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47.

Seroussi, Y., Bohnert, F., and Zukerman, I. (2012). Authorship attribution with author-aware topic models. In *ACL (2)*, pages 264–269. The Association for Computer Linguistics.

Shazeer, N., Doherty, R., Evans, C., and Waterson, C. (2016). Swivel: Improving embeddings by noticing what's missing. *CoRR*, abs/1602.02215.

Shrestha, P., Sierra, S., Gonzalez, F., Montes, M., Rosso, P., and Solorio, T. (2017). Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674. Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Stamatatos, E. (2009). A survey of modern authorship at-

tribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Stamatatos, E. (2017). Authorship attribution using text distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1138–1149, Valencia, Spain, April. Association for Computational Linguistics.

Tanguy, L., Sajous, F., Calderone, B., and Hathout, N. (2012). Authorship Attribution: Using Rich Linguistic Features when Training Data is Scarce. In *PAN Lab at CLEF*, pages –, Rome, Italy, September.