2018 21st International Conference of Computer and Information Technology (ICCIT), 21-23 December, 2018

A Comparative Analysis of Word Embedding Representations in Authorship Attribution of Bengali Literature

Hemayet Ahmed Chowdhury
Department of Computer Science and Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
Email: hemayetchoudhury@gmail.com

Md. Azizul Haque Imon
Department of Computer Science and Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
Email: azizulhaqueimon@gmail.com

Md. Saiful Islam
Department of Computer Science and Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
Email: saif.acm@gmail.com

***Abstract*—Word Embeddings can be used by deep layers of neural networks to extract features from them to learn stylometric patterns of authors based on context and co-occurrence of the words in the field of Authorship Attribution. In this paper, we investigate the effects of different types of word embeddings in Authorship Attribution of Bengali Literature, specifically the skip-gram and continuous-bag-of-words(CBOW) models generated by Word2Vec and fastText along with the word vectors generated by Glove . We experiment with dense neural network models, such as the convolutional and recurrent neural networks and analyse how different word embedding models effect the performance of the classifiers and discuss their properties in this classification task of Authorship Attribution of Bengali Literature. The experiments are performed on a data set we prepared, consisting of 2400 on-line blog articles from 6 authors of recent times.**

***Keywords*—word embeddings, bengali literature, authorship attribution, skip-gram, continuous-bag-of-words, fastText, Word2Vec, Glove, deep learning, convolutional neural network, recurrent neural network**

I. Introduction

Authorship Attribution is the task of identifying the original author of a given piece of text, by analyzing previous of works of the authors in question. In traditional methods of authorship attribution, independent features such as lexical n-grams or frequency based word embeddings are used to represent the text, which are very similar to one-hot encodings. As well as the methods perform, in such approaches, the word representations are created independent of each other's meanings and words of similar contexts seem to be represented in different vector spaces, which is problematic for detecting the semantic values of words. Word embeddings, also generally known as distributed term representations, take a different approach in creating representations such that each term is encoded as low-dimensional dense vectors which are not discrete, but continuous on contrast to the one-hot representations. This approach allows the word embeddings to encode semantic and syntactic similarity between words by capturing the extent to which words appear in contexts that are similar [8]. Word2Vec, one of the most popular set of algorithms used for implementing word embeddings in modern times was proposed by Mikolov and Dean[14]. Other technologies like Glove and Facebook's fastText can also be used for word embeddings[11]. Two common variants offered by Word2Vec and fastText are the skip-gram model, which predicts the surrounding words given a target word and the continuous bag-of-words(CBOW), which predicts the target word, given the context of the neighboring words. The Glove model varies slightly in this aspect, such that it uses a method of maximizing the probability in alignment of words and statistics based on co-occurrence to generate word embeddings.

Due to their ability to hold context and semantic meanings of words, our approach in this paper was to investigate how word embeddings perform in the task of Authorship Attribution in Bengali. Here, we will discuss different types of embeddings, their properties, advantages and disadvantages. We will also present analysis of how the performance rates of different neural network classifiers react to the different variants of word embeddings. No work, analysis or investigation has been published on the effect of word embeddings in Authorship Attribution of Bengali Literature as of our knowledge to date.

II. Related Works

A. On Author Attribution

Studies regarding author attribution has been going on for quite some time now. Initial works had Mosteller and Wallace working with distribution of 30 function words comprising of conjunctions, prepositions and articles on federalist papers to attribute to the original authors[15].Bogdanova and Lazaridou conducted experiments with cross-language Authorship Attribution, using books from 6 English authors along with their Spanish translations and eventually proposed that Machine

978-1-5386-9242-4/18/$31.00

Translation could be used as a starting point for cross-language Authorship Attribution[1]. Nasir et al.[16] approached Authorship Attribution as semi-supervised anomaly detection via multiple kernel learning, whereas Zhao et al.[25] used Kullback-Leibler divergence with Dirichlet smoothing on AP, Gutenberg, and Reuters-21578 corpora to obtain impressive results. Sanderson and Guenter [2006] studied character and word sequence kernels for authorship attribution of short texts. Two Markov chain approaches were used to compare the performances[21]. Application of several configurations of sequence kernels on a multi-topic dataset of 50 authors resulted in better performance of character sequence kernels than word sequence kernels. These observations suggested that amount of training data had more of an impact on discrimination power than size of text data.In 2007 Jonathan H. Clark attempted authorship attribution with the use of synonym-based features for their experiment[4] where as,Bozkurt I. N. et Al. took a different but very effective approach with stylometry along with features like Vocabulary Diversity, Bag of Words and Frequency of Function words (article, pronoun, conjunction) to identify writing characteristics of five Milliyet columnists[2].

Compared to the progress in research regarding authorship attribution in English and German literature, such research work for Bangla has not yet set a high benchmark. Only three notable research works can be identified for Bangla.Das and Mitra studied authorship attribution and worked on a data set of three authors consisting of a total of 36 documents[7].They used uni-gram and bi-gram features along with a probabilistic classification method.The uni-gram yielded 90% accuracy while the bi-gram yielded a staggering 100% accuracy.However, their data set was small and the authors had very different styles of writing, which made it easier for classification.

Chakraborty performed a ten-fold cross-validation on three classes and showed that SVM classifiers can provide best accuracy of up to 84%[3].Jana looked into Sister Niveditas influence on Jagadish Chandra Boses writings[10] but no classification was performed. Other than these three works, Shanta Phani also attempted attribution on three authors using machine learning techniques, much similar to Suprabhat Dass work[19].
P. Das, R. Tasmim, and S. Ismail have performed experiments on four Bangladeshi authors of current time using features like word frequency , word and sentence length, type-token ratio, number of conjunction and pronoun, etc[6]. Hossain and Rahman developed a voting system with multiple features classified with Cosine similarity ,achieving an accuracy of 90.67%[9]. Pal, Siddika and Ismail achieved an accuracy of 90.74% based on 6 authors with a support vector machine on a single feature[17].None of these works crossed a mark of 90% accuracy satisfactorily.
Excluding the work with multilayered perceptrons by Phani, Lahiri and Biswas[18], we did not find much work done with neural networks, and absolutely none at all with LSTM or convolutional neural nets or word embeddings.

B. On Word Embeddings

Tripodi et al. [23] analyzed the performances of CBOW and Skip-gram algorithms for Italian language by tuning different hyper parameter. Vine et al. [13] investigated the use of unsupervised features which were derived from the word embedding approaches and found results that indicates that the use of word embeddings improve the effectiveness of concept extraction method. In 2017,Haixia Liu attempted citation sentiment analysis using Word2Vec and found that word embeddings are effective for classifying positive and negative citation[12]. Convolutional neural network with word embeddings as it's feature was used by Santos et al. [22] which concluded that both fastText and Word2Vec outperform the baseline models like Support Vector Machine, Random Forest, Logistic Regression etc. Rudkowsky et al. [20] found that word embeddings have potential to improve on current bag-of-words approaches in the field of sentiment analysis in the social sciences.Joulin et al. [11] found that fastText classifier gives a high accuracy which is on par with deep learning classifiers and also faster for training and evaluation.

III. Methodology

A. Corpus

The dataset that yielded high accuracies for algorithms, such as the voting system and cosine similarity[9], was our primary target for implementing our training models. Further analysis showed that high accuracies were only achieved for the dataset upon implementation of shallow models, such as the Support Vector Machine (SVM) and Naive Bayes(NB). It was basically due to the limited size and lack of sparsity of the dataset. Following such observations, we developed a custom web parser to accumulate our own dataset and another parser to normalize the collected stream of raw data. Due to privacy and copyright issues, the authors have been code-named as FE, HM, EJ, MT, RN and RG in this paper. Table 1 illustrates total size of the corpus in words, for each author.

Author	Total Words
FE	2382241
HM	3372688
EJ	2401315
MT	2515428
RN	1418598
RG	2454574

TABLE I
Corpus Size Per Author

From table 1 and Figure 1, we can say that the corpus is moderately balanced. This is due to the fact that all the documents were collected from online blogs, which makes the corpus realistic and sparse.

B. Word Embeddings

A word embedding format basically tries to store words in a vector space as a vector representation. In practice, this usually means that word embeddings are placed in a high dimensional space where the embeddings of similar or related words

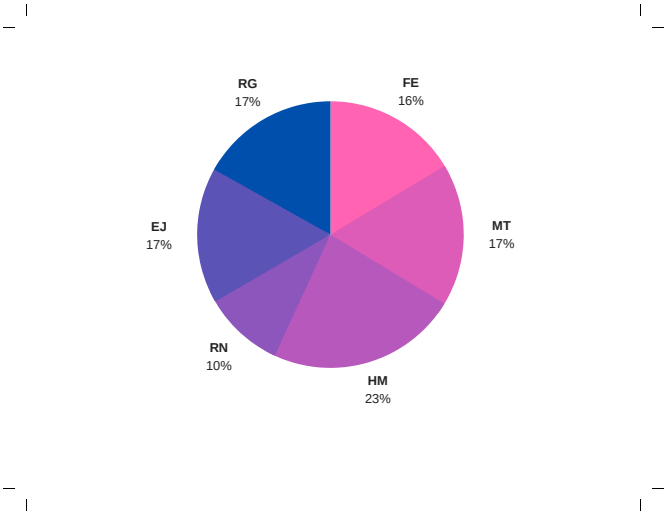

Fig. 1. Word Distribution Per Author

Input	Output		Hey	This	is	sample	corpus	using	only	one	context	word
Hey	this	Datapoint 1	1	0	0	0	0	0	0	0	0	0
this	hey	Datapoint 2	0	1	0	0	0	0	0	0	0	0
is	this	Datapoint 3	0	0	1	0	0	0	0	0	0	0
is	sample	Datapoint 4	0	0	1	0	0	0	0	0	0	0
sample	is	Datapoint 5	0	0	0	1	0	0	0	0	0	0
sample	corpus	Datapoint 6	0	0	0	1	0	0	0	0	0	0
corpus	sample	Datapoint 7	0	0	0	0	1	0	0	0	0	0
corpus	using	Datapoint 8	0	0	0	0	1	0	0	0	0	0
using	corpus	Datapoint 9	0	0	0	0	0	1	0	0	0	0
using	only	Datapoint 10	0	0	0	0	0	1	0	0	0	0
only	using	Datapoint 11	0	0	0	0	0	0	1	0	0	0
only	one	Datapoint 12	0	0	0	0	0	0	1	0	0	0
one	only	Datapoint 13	0	0	0	0	0	0	0	1	0	0
one	context	Datapoint 14	0	0	0	0	0	0	0	1	0	0
context	one	Datapoint 15	0	0	0	0	0	0	0	0	1	0
context	word	Datapoint 16	0	0	0	0	0	0	0	0	1	0
word	context	Datapoint 17	0	0	0	0	0	0	0	0	0	1

Fig. 2. CBOW training set configuration

Hey	this	is	sample	corpus	using	only	one	context	word
0	0	0	1	0	0	0	0	0	0

Fig. 3. CBOW target

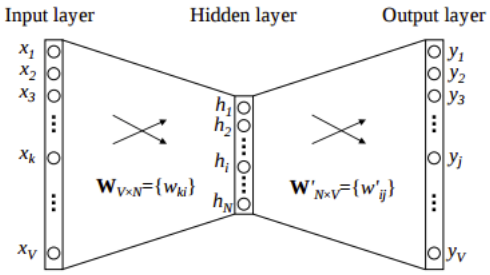

Fig. 4. CBOW architecture for single word

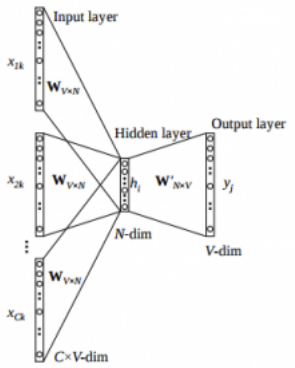

Fig. 5. CBOW architecture for multi-words

are close to each other and different word embeddings are placed far from each other.Since, Machine learning including deep learning cannot process strings, p lain texts and require vectorized version of the texts for operations, word embedding was the preferred approach. Word embedding is generally classified into two classes, which are:

- Frequency based embedding
- Prediction based embedding

We opted to go with the prediction based embedding. But, such word embedding methods had limitations until Mikolov et. al.[14] introduced word2vec to the NLP society. A task like King man +woman= Queen was achieved, based on his introduced prediction based methods. Further work on these methods resulted in more efficient frameworks such as GloVe and FastText.

Generally, prediction based word embeddings can be seen in two different flavours - CBOW and Skip-gram. But, the GloVe framework mentioned above uses a third type of technology.

1) Continuous Bag of Words: The CBOW model tries to predict the probability of a word, according to its context. Representation of the context is like a bag of the contained words, in a fixed sized window, around the target word. The example below will illustrate this concept further. If a corpus C represents the text "Hey, this is a sample corpus using only one context word, given the context window is set to be 1, the corpus can be configured in the following way to be a training set:

So, the target for the data point will appear a lot like this:

The single word architecture of a CBOW model is given below for better understanding:

The base concept of multi-word architecture remains the same but the architecture is a bit more complex. The diagram of multi-word CBOW model architecture is presented below:

Hence, CBOW basically is like predicting a word if the context is given.

2) Skip-Gram: The Skip-Gram architecture is actually an inverted architecture of CBOWs. On contrary to the CBOW model, the skip gram model is basically like predicting the context if a word is given. Here, more distant words are given less weight by randomly sampling them.While defining the window size parameter, only the maximum window size can be configured. Actual window size varies from 1 to max size and are chosen at random.

If a context window size of 1 is chosen for both CBOW and skip-gram models, two one hot encoded target variables will act as the targets along with two corresponding outputs. Two error vectors which will be achieved by calculating two different errors with corresponding to two target variables,

will be added element-wise to get a final error vector. After training, the weights between the input and the hidden layer will be considered as the word vector representation . The loss function or the objective is pretty much the same type as of the CBOW model. The architecture of Skip-Gram model lies below:

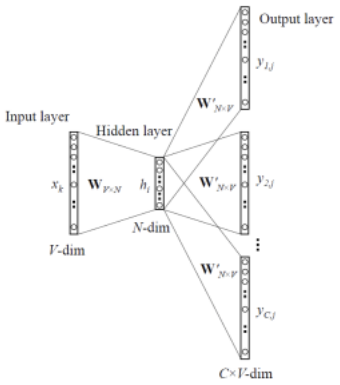

Fig. 6. Skip-gram architecture

3) GloVe: GloVE has quite a similar working mechanism to the Word2Vec model mentioned above. Though word2vec predicts context given words, GLOVE learns by constructing a co-occurrence matrix, that basically counts how frequently a word appears in a context. Since the matrix becomes gigantic, we factorize this matrix for a lower-dimension representation. When modeling the loss function, the glove model basically tries to use the probability ratio of the words that appear in the context.

For our analysis purpose, we used the following embeddings for training on our dataset, each consisting of 100 dimensions:

- CBOW model by Word2Vec
- Skip-Gram model by Word2Vec
- CBOW model by fastText
- Skip-Gram model by fastText
- Word Embeddings by GloVe

In generating the word level embeddings based on the 5 models mentioned above, we used the following parameters for all 5 of the models, for fair experimentation.

- Vector Dimensions of length 100.
- Context window of 5.
- 5 learning iterations.

C. Architectures of Proposed Models

Deep learning is a specialized branch of machine learning which can learn features directly from the given datasets, without having any prior knowledge of the features. Generally, deep learning approaches are one of the methods for learning patterns from word embeddings. On the held out dataset, we implemented customized architectures of the following deep learning models, with word embeddings as their input, to obtain an analysis of how different word embeddings perform with different types of neural networks.

- A Convolutional Neural Network.(CNN)
- A Classical Multi-layered Perceptron.(NN)
- A Recurrent Neural Network with Long Short Term Memory.(RNN with LSTM)

1) The Convolutional Neural Network Model: The CNN architecture we use is a modification of the one used by Collobert et al. [5]. Our model takes as input, a text which is padded to a length of 2000 words. The CNN is based on a single channel, where we represent the text as a concatenation of its word embeddings generated by the different algorithms previously mentioned. The convolutional layer slides a filter of window size 64 over the input channel. The filter, in turn, generates a new feature for a window of words. The application of the filter over each possible window of words in the sentence produces a feature map. Max-over-time pooling [5] in turn condenses this feature vector to its most important feature by taking its maximum value and naturally deals with variable input lengths. A final softmax layer takes the concatenation of the maximum values of the feature maps produced by all filters and outputs a probability distribution over all candidate authors.

Collobert et al. [5] used a CNN with a non-static word embedding channel where the vectors are modified during training using backpropagation. Our approach varies in the sense that the word embedding channel is kept static and the vectors that are pre-trained by algorithms such the Word2Vec, fastText and Glove are not modified during training.

2) The Multi-layered Perceptron model: The multi-layered perceptron model is also similar to that of the CNN architecture we used. It takes the concatenation of the word embeddings as input to the first layer. 2 more dense layers with 'RELU' activation function are added to extract the most important features and patterns from the word embeddings. A final softmax layer outputs the probability distribution of the authors present in the data set. The pre-trained word embeddings are not modified during training.

3) The RNN model: The Recurrent Neural Network architecture we proposed uses LSTM gates. The word embedding representations of the padded texts are the input for a series of non-linear operations in the LSTM layer [24]. Similar to the MLP architecture, 2 more hidden layers are used to extract the features from the representations. Again, a final softmax layer outputs the probability distribution of the authors, while the pre-trained word vectors are kept unmodified.

IV. Experiments and Result Analysis

To analyze the performance of the word embeddings, the models were trained on 2100 articles, and tested on the other 300. We settled on a testing data set size of 12.5% instead of opting for the 80:20 ratio due to the moderate size of the data set. Increasing the testing data size in this case would significantly hamper the training our models would receive.

In the following figure, the performance of different word embeddings is shown using different classifiers specifically the NN, RNN with LSTM and CNN.

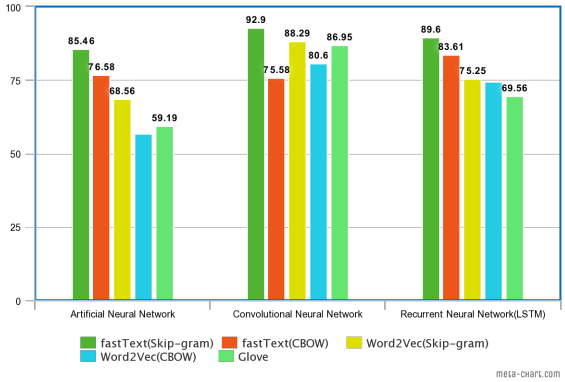

Fig. 7. Performance Analysis of Different Feature Sets with Deep Neural Networks

Classifier	Representation	Accuracy
ANN	fastText(skip-gram)	85.46%
	fastText(CBOW)	76.58%
	W2V(skip-gram)	68.56%
	W2V(CBOW)	56.85%
	Glove	59.19%
RNN	fastText(skip-gram)	89.6%
	fastText(CBOW)	83.61%
	W2V(skip-gram)	75.25%
	W2V(CBOW)	74.58%
	Glove	69.56%
CNN	fastText(skip-gram)	92.9%
	fastText(CBOW)	75.58%
	W2V(skip-gram)	88.29%
	W2V(CBOW)	80.60%
	Glove	86.95%

TABLE II
PERFORMANCE ANALYSIS OF DIFFERENT FEATURE SETS WITH DEEP NEURAL NETWORKS

From Figure 7 and Table 2, we can see that the convolutional neural networks seem to work best when combined with word-embeddings on the held out data set.

The most important observation of this study, however, is seen in the diagrammatic figures, where the skip-gram models tend to outperform the CBOW and embeddings by Glove. Since the task was authorship attribution, the skip gram models' nature of trying to predict the context from words seemed to work best in this case. The skip-gram word embeddings by fastText seem to outperform the word representations by the other algorithms on almost all models.

V. CONCLUSION

As of date, no research has been published investigating the effects of word embeddings with deep neural networks for authorship attribution in Bengali. Word embeddings can hold semantic values of bengali terms, which is a key attribute in the field of authorship attribution in this language. Our contribution in this paper was exploring the use of word embeddings for authorship attribution and providing a comparison between the different word representation algorithms, to show which works best in Bengali Language.Wwe discussed the different types of word embeddings in depth, along with their performance analysis when combined with different neural network models and came to the conclusion that the skip-gram word embeddings by fastText tend to perform better than embeddings by Word2Vec or Glove in this specific task. For future work, we would like to investigate character level embeddings by different algorithms and how they compare to word level embeddings in the task of Authorship Attribution in Bengali Language.

REFERENCES

[1] Dasha Bogdanova and Angeliki Lazaridou. "Cross-language authorship attribution". In: *Proceedings of the 9th International Conference on Language Resources and Evaluation* (2014).

[2] I. N. Bozkurt, O. Baglioglu, and E. Uyar. "Authorship attribution". In: *2007 22nd international symposium on computer and information sciences* (Nov. 2007), pp. 1–5.

[3] Tanmoy Chakraborty. "Authorship Identification Using Stylometry Analysis in Bengali Literature". In: *CoRR* (2012).

[4] Jonathan H. Clark and Charles J. Hannon. "A Classifier System for Author Recognition Using Synonym-Based Features". In: *MICAI 2007: Advances in Artificial Intelligence*. 2007, pp. 839–849.

[5] Ronan Collobert et al. "Natural Language Processing (Almost) from Scratch". In: *J. Mach. Learn. Res.* 12 (Nov. 2011), pp. 2493–2537. ISSN: 1532-4435.

[6] P. Das, R. Tasmim, and S. Ismail. "An experimental study of stylometry in bangla literature". In: *Electrical Information and Communication Technology (EICT), 2015 2nd International Conference* (2015).

[7] Suprabhat Das and Pabitra Mitra. "Author identification in Bengali literary works". In: *Pattern Recognition and Machine Intelligence,* (2011).

[8] J. Firth. "A Synopsis of Linguistic Theory 1930-1955". In: *Studies in Linguistic Analysis*. reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow. Philological Society, Oxford, 1957.

[9] M. Tahmid Hossain et al. "A stylometric analysis on Bengali literature for authorship attribution". In: *Computer and Information Technology (ICCIT), 2017 20th International Conference of IEEE* (2017).

[10] Siladitya Jana. "Sister Nivedita's influence on J. C. Bose's writings". In: (2015).

[11] Armand Joulin et al. "Bag of tricks for efficient text classification". In: *arXiv preprint arXiv:1607.01759* (2016).

[12] Haixia Liu. "Sentiment analysis of citations using word2vec". In: *arXiv preprint arXiv:1704.00177* (2017).

[13] and Mahnoosh Kholghi et al. “Analysis of Word Embeddings and Sequence Features for Clinical Information Extraction”. In: *Proceedings of the Australasian Language Technology Association Workshop 2015*. Parramatta, Australia, 2015, pp. 21–30.

[14] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *CoRR* abs/1301.3781 (2013).

[15] Frederick Mosteller and David L. Wallace. “Inference in an Authorship Problem”. In: *Journal of the American Statistical Association* 58.302 (1963), pp. 275–309.

[16] A. Jamal Nasir, Nico Gornitz, and Ulf Brefeld. “An off-the-shelf approach to authorship attribution”. In: (2014).

[17] U. Pal, A. S. Nipu, and S. Ismail. “A machine learning approach for stylometric analysis of Bangla literature”. In: *2017 20th International Conference of Computer and Information Technology (ICCIT)*. Dec. 2017, pp. 1–5.

[18] S. Phani, S. Lahiri, and A. Biswas. “A machine learning approach for authorship attribution for Bengali blogs”. In: (Nov. 2016), pp. 271–274.

[19] S. Phani et al. “Authorship attribution in bengali language”. In: ().

[20] Elena Rudkowsky et al. “More than Bags of Words: Sentiment Analysis with Word Embeddings”. In: *Communication Methods and Measures* 12.2-3 (2018), pp. 140–157.

[21] Conrad Sanderson and Simon Guenter. “Short Text Authorship Attribution via Sequence Kernels, Markov Chains and Author Unmasking: An Investigation”. In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. EMNLP ’06. Sydney, Australia, 2006, pp. 482–491.

[22] Igor Santos, Nadia Nedjah, and Luiza de Macedo Mourelle. “Sentiment analysis using convolutional neural network with fastText embeddings”. In: *Computational Intelligence (LA-CCI), 2017 IEEE Latin American Conference on*. IEEE. 2017, pp. 1–5.

[23] Rocco Tripodi and Stefano Li Pira. “Analysis of Italian Word Embeddings”. In: *arXiv preprint arXiv:1707.08783* (2017).

[24] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. “Recurrent Neural Network Regularization”. In: *CoRR* abs/1409.2329 (2014). arXiv: 1409.2329. URL: http://arxiv.org/abs/1409.2329.

[25] Ying Zhao, Justin Zobel, and Phil Vines. “Using Relative Entropy for Authorship Attribution”. In: *Information Retrieval Technology*. 2006, pp. 92–105.