Automated essay scoring with string kernels and word embeddings

Mădălina Cozma and **Andrei M. Butnaru** and **Radu Tudor Ionescu**
University of Bucharest
Department of Computer Science
14 Academiei, Bucharest, Romania
`butnaruandreimadalin@gmail.com`
`raducu.ionescu@gmail.com`

Abstract

In this work, we present an approach based on combining string kernels and word embeddings for automatic essay scoring. String kernels capture the similarity among strings based on counting common character n-grams, which are a low-level yet powerful type of feature, demonstrating state-of-the-art results in various text classification tasks such as Arabic dialect identification or native language identification. To our best knowledge, we are the first to apply string kernels to automatically score essays. We are also the first to combine them with a high-level semantic feature representation, namely the bag-of-super-word-embeddings. We report the best performance on the Automated Student Assessment Prize data set, in both in-domain and cross-domain settings, surpassing recent state-of-the-art deep learning approaches.

1 Introduction

Automatic essay scoring (AES) is the task of assigning grades to essays written in an educational setting, using a computer-based system with natural language processing capabilities. The aim of designing such systems is to reduce the involvement of human graders as far as possible. AES is a challenging task as it relies on grammar as well as semantics, pragmatics and discourse (Song et al., 2017). Although traditional AES methods typically rely on handcrafted features (Larkey, 1998; Foltz et al., 1999; Attali and Burstein, 2006; Dikli, 2006; Wang and Brown, 2008; Chen and He, 2013; Somasundaran et al., 2014; Yannakoudakis et al., 2014; Phandi et al., 2015), recent results indicate that state-of-the-art deep learning methods reach better performance (Alikaniotis et al., 2016; Dong and Zhang, 2016; Taghipour and Ng, 2016; Dong et al., 2017; Song et al., 2017; Tay et al., 2018), perhaps because these methods are able to

capture subtle and complex information that is relevant to the task (Dong and Zhang, 2016).

In this paper, we propose to combine string kernels (low-level character n-gram features) and word embeddings (high-level semantic features) to obtain state-of-the-art AES results. Since recent methods based on string kernels have demonstrated remarkable performance in various text classification tasks ranging from authorship identification (Popescu and Grozea, 2012) and sentiment analysis (Giménez-Pérez et al., 2017; Popescu et al., 2017) to native language identification (Popescu and Ionescu, 2013; Ionescu et al., 2014; Ionescu, 2015; Ionescu et al., 2016; Ionescu and Popescu, 2017) and dialect identification (Ionescu and Popescu, 2016; Ionescu and Butnaru, 2017), we believe that string kernels can reach equally good results in AES. To the best of our knowledge, string kernels have never been used for this task. As string kernels are a simple approach that relies solely on character n-grams as features, it is fairly obvious that such an approach will not to cover several aspects (e.g.: semantics, discourse) required for the AES task. To solve this problem, we propose to combine string kernels with a recent approach based on word embeddings, namely the bag-of-super-word-embeddings (BOSWE) (Butnaru and Ionescu, 2017). To our knowledge, this is the first successful attempt to combine string kernels and word embeddings. We evaluate our approach on the Automated Student Assessment Prize data set, in both in-domain and cross-domain settings. The empirical results indicate that our approach yields a better performance than state-of-the-art approaches (Phandi et al., 2015; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018).

2 Method

String kernels. Kernel functions (Shawe-Taylor and Cristianini, 2004) capture the intuitive notion of similarity between objects in a specific do-

main. For example, in text mining, string kernels can be used to measure the pairwise similarity between text samples, simply based on character n-grams. Various string kernel functions have been proposed to date (Lodhi et al., 2002; Shawe-Taylor and Cristianini, 2004; Ionescu et al., 2014). One of the most recent string kernels is the *histogram intersection string kernel* (HISK) (Ionescu et al., 2014). For two strings over an alphabet Σ, $x, y \in \Sigma^*$, the intersection string kernel is formally defined as follows:

$$k^{\cap}(x, y) = \sum_{v \in \Sigma^n} \min\{\text{num}_v(x), \text{num}_v(y)\}, \quad (1)$$

where $\text{num}_v(x)$ is the number of occurrences of n-gram v as a substring in x, and n is the length of v. In our AES experiments, we use the intersection string kernel based on a range of character n-grams. We approach AES as a regression task, and employ ν-Support Vector Regression (ν-SVR) (Suykens and Vandewalle, 1999; Shawe-Taylor and Cristianini, 2004) for training.

Bag-of-super-word-embeddings. Word embeddings are long known in the NLP community (Bengio et al., 2003; Collobert and Weston, 2008), but they have recently become more popular due to the *word2vec* (Mikolov et al., 2013) framework that enables the building of efficient vector representations from words. On top of the word embeddings, Butnaru and Ionescu (2017) developed an approach termed *bag-of-super-word-embeddings* (BOSWE) by adapting an efficient computer vision technique, the bag-of-visual-words model (Csurka et al., 2004), for natural language processing tasks. The adaptation consists of replacing the image descriptors (Lowe, 2004) useful for recognizing object patterns in images with word embeddings (Mikolov et al., 2013) useful for recognizing semantic patterns in text documents.

The BOSWE representation is computed as follows. First, each word in the collection of training documents is represented as word vector using a pre-trained word embeddings model. Based on the fact that word embeddings carry semantic information by projecting semantically related words in the same region of the embedding space, the next step is to cluster the word vectors in order to obtain relevant semantic clusters of words. As in the standard bag-of-visual-words model, the clustering is done by k-means (Leung and Malik, 2001), and the formed centroids are stored in a randomized forest of k-d trees (Philbin et al., 2007) to re-duce search cost. The centroid of each cluster is interpreted as a *super word embedding* or *super word vector* that embodies all the semantically related word vectors in a small region of the embedding space. Every embedded word in the collection of documents is then assigned to the nearest cluster centroid (the nearest super word vector). Put together, the super word vectors generate a vocabulary (codebook) that can further be used to describe each document as a *bag-of-super-word-embeddings*. To obtain the BOSWE represenation for a document, we just have to compute the occurrence count of each super word embedding in the respective document. After building the representation, we employ a kernel method to train the BOSWE model for our specific task. To be consistent with the string kernel approach, we choose the histogram intersection kernel and the same regression method, namely ν-SVR.

Model fusion. In the primal form, a linear classifier takes as input a feature matrix X of r samples (rows) with m features (columns) and optimizes a set of weights in order to reproduce the r training labels. In the dual form, the linear classifier takes as input a kernel matrix K of $r \times r$ components, where each component k_{ij} is the similarity between examples x_i and x_j. Kernel methods work by embedding the data in a Hilbert space and by searching for linear relations in that space, using a learning algorithm. The embedding can be performed either (i) implicitly, by directly specifying the similarity function between each pair of samples, or (ii) explicitly, by first giving the embedding map ϕ and by computing the inner product between each pair of samples embedded in the Hilbert space. For the linear kernel, the associated embedding map is $\phi(x) = x$ and options (i) or (ii) are equivalent, i.e. the similarity function is the inner product. Hence, the linear kernel matrix K can be obtained as $K = X \cdot X'$, where X' is the transpose of X. For other kernels, e.g. the histogram intersection kernel, it is not possible to explicitly define the embedding map (Shawe-Taylor and Cristianini, 2004), and the only solution is to adopt option (i) and compute the corresponding kernel matrix directly. Therefore, we combine HISK and BOSWE in the dual (kernel) form, by simply summing up the two corresponding kernel matrices. However, summing up kernel matrices is equivalent to feature vector concatenation in the primal Hilbert space. To better explain

Prompt	Number of Essays	Score Range
1	1783	2-12
2	1800	1-6
3	1726	0-3
4	1726	0-3
5	1772	0-4
6	1805	0-4
6	1569	0-30
6	723	0-60

Table 1: The number of essays and the score ranges for the 8 different prompts in the Automated Student Assessment Prize (ASAP) data set.

this statement, let us suppose that we can define the embedding map of the histogram intersection kernel and, consequently, we can obtain the corresponding feature matrix of HISK with $r \times m_1$ components denoted by X_1 and the corresponding feature matrix of BOSWE with $r \times m_2$ components denoted by X_2. We can now combine HISK and BOSWE in two ways. One way is to compute the corresponding kernel matrices $K_1 = X_1 \cdot X_1'$ and $K_2 = X_2 \cdot X_2'$, and to sum the matrices into a single kernel matrix $K_+ = K_1 + K_2$. The other way is to first concatenate the feature matrices into a single feature matrix $X_+ = [X_1 X_2]$ of $r \times (m_1 + m_2)$ components, and to compute the final kernel matrix using the inner product, i.e. $K_+ = X_+ \cdot X_+'$. Either way, the two approaches, HISK and BOSWE, are fused before the learning stage. As a consequence of kernel summation, the search space of linear patterns grows, which should help the kernel classifier, in our case ν-SVR, to find a better regression function.

3 Experiments

Data set. To evaluate our approach, we use the Automated Student Assessment Prize (ASAP) [1] data set from Kaggle. The ASAP data set contains 8 prompts of different genres. The number of essays per prompt along with the score ranges are presented in Table 1. Since the official test data of the ASAP competition is not released to the public, we, as well as others before us (Phandi et al., 2015; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018), use only the training data in our experiments.

Evaluation procedure. As Dong and Zhang (2016), we scaled the essay scores into the range

---

[1] https://www.kaggle.com/c/asap-aes/data

0-1. We closely followed the same settings for data preparation as (Phandi et al., 2015; Dong and Zhang, 2016). For the in-domain experiments, we use 5-fold cross-validation. The 5-fold cross-validation procedure is repeated for 10 times and the results were averaged to reduce the accuracy variation introduced by randomly selecting the folds. We note that the standard deviation in all cases in below 0.2%.

For the cross-domain experiments, we use the same source→target domain pairs as (Phandi et al., 2015; Dong and Zhang, 2016), namely, 1→2, 3→4, 5→6 and 7→8. All essays in the source domain are used as training data. Target domain samples are randomly divided into 5 folds, where one fold is used as test data, and the other 4 folds are collected together to sub-sample target domain train data. The sub-sample sizes are $n_t = \{10, 25, 50, 100\}$. The sub-sampling is repeated for 5 times as in (Phandi et al., 2015; Dong and Zhang, 2016) to reduce bias. As our approach performs very well in the cross-domain setting, we also present experiments *without* sub-sampling data from the target domain, i.e. when the sub-sample size is $n_t = 0$. As evaluation metric, we use the quadratic weighted kappa (QWK).

Baselines. We compare our approach with state-of-the-art methods based on handcrafted features (Phandi et al., 2015), as well as deep features (Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018). We note that results for the cross-domain setting are reported only in some of these recent works (Phandi et al., 2015; Dong and Zhang, 2016).

Implementation choices. For the string kernels approach, we used the histogram intersection string kernel (HISK) based on the blended range of character n-grams from 1 to 15. To compute the intersection string kernel, we used the open-source code provided by Ionescu et al. (2014). For the BOSWE approach, we used the pre-trained word embeddings computed by the *word2vec* toolkit (Mikolov et al., 2013) on the Google News data set using the Skip-gram model, which produces 300-dimensional vectors for 3 million words and phrases. We used functions from the VLFeat library (Vedaldi and Fulkerson, 2008) for the other steps involved in the BOSWE approach, such as the k-means clustering and the randomized forest of k-d trees. We set the number of clusters (dimension of the vocabulary) to $k = 500$. After

Method	1	2	3	4	5	6	7	8	Overall
Human	0.721	0.814	0.769	0.851	0.753	0.776	0.721	0.629	0.754
(Phandi et al., 2015)	0.761	0.606	0.621	0.742	0.784	0.775	0.730	0.617	0.705
(Dong and Zhang, 2016)	-	-	-	-	-	-	-	-	0.734
(Dong et al., 2017)	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
(Tay et al., 2018)	0.832	0.684	**0.695**	0.788	0.815	0.810	0.800	0.697	0.764
HISK and ν-SVR	0.836	0.724	0.677	0.821	0.830	0.828	0.801	0.726	0.780
BOSWE and ν-SVR	0.788	0.689	0.667	0.809	0.824	0.824	0.766	0.679	0.756
HISK+BOSWE and ν-SVR	**0.845**	**0.729**	0.684	**0.829**	**0.833**	**0.830**	**0.804**	**0.729**	**0.785**

Table 2: In-domain automatic essay scoring results of our approach versus several state-of-the-art methods (Phandi et al., 2015; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018). Results are reported in terms of the quadratic weighted kappa (QWK) measure, using 5-fold cross-validation. The best QWK score (among the machine learning systems) for each prompt is highlighted in bold.

computing the BOSWE representation, we apply the L_1-normalized intersection kernel. We combine HISK and BOSWE in the dual form by summing up the two corresponding matrices. For the learning phase, we employ the dual implementation of ν-SVR available in LibSVM (Chang and Lin, 2011). We set its regularization parameter to $c = 10^3$ and $\nu = 10^{-1}$ in all our experiments.

In-domain results. The results for the in-domain automatic essay scoring task are presented in Table 2. In our empirical study, we also include feature ablation results. We report the QWK measure on each prompt as well as the overall average. We first note that the histogram intersection string kernel alone reaches better overall performance (0.780) than all previous works (Phandi et al., 2015; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018). Remarkably, the overall performance of the HISK is also higher than the inter-human agreement (0.754). Although the BOSWE model can be regarded as a shallow approach, its overall results are comparable to those of deep learning approaches (Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018). When we combine the two models (HISK and BOSWE), we obtain even better results. Indeed, the combination of string kernels and word embeddings attains the best performance on 7 out of 8 prompts. The average QWK score of HISK and BOSWE (0.785) is more than 2% better the average scores of the best-performing state-of-the-art approaches (Dong et al., 2017; Tay et al., 2018).

Cross-domain results. The results for the cross-domain automatic essay scoring task are presented in Table 3. For each and every source→target pair, we report better results than both state-of-the-art

methods (Phandi et al., 2015; Dong and Zhang, 2016). We observe that the difference between our best QWK scores and the other approaches are sometimes much higher in the cross-domain setting than in the in-domain setting. We particularly notice that the difference from (Phandi et al., 2015) when $n_t = 0$ is always higher than 10%. Our highest improvement (more than 54%, from 0.187 to 0.728) over (Phandi et al., 2015) is recorded for the pair 5→6, when $n_t = 0$. Our score in this case (0.728) is even higher than both scores of Phandi et al. (2015) and Dong and Zhang (2016) when they use $n_t = 50$. Different from the in-domain setting, we note that the combination of string kernels and word embeddings does not always provide better results than string kernels alone, particularly when the number of target samples (n_t) added into the training set is less or equal to 25.

Discussion. It is worth noting that in a set of preliminary experiments (not included in the paper), we actually considered another approach based on word embeddings. We tried to obtain a document embedding by averaging the word vectors for each document. We computed the average as well as the standard deviation for each component of the word vectors, resulting in a total of 600 features, since the word vectors are 300-dimensional. We applied this method in the in-domain setting and we obtained a surprisingly low overall QWK score, around 0.251. We concluded that this simple approach is not useful, and decided to use BOSWE (Butnaru and Ionescu, 2017) instead.

It would have been interesting to present an error analysis based on the discriminant features weighted higher by the ν-SVR method. Unfortu-

Source→Target	Method	$n_t = 0$	$n_t = 10$	$n_t = 25$	$n_t = 50$	$n_t = 100$
1→2	(Phandi et al., 2015)	0.434	0.463	0.457	0.492	0.510
	(Dong and Zhang, 2016)	-	0.546	0.569	0.563	0.559
	HISK and ν-SVR	0.440	**0.586**	**0.637**	0.652	0.657
	BOSWE and ν-SVR	0.398	0.474	0.478	0.492	0.506
	HISK+BOSWE and ν-SVR	**0.542**	0.584	0.632	**0.657**	**0.661**
3→4	(Phandi et al., 2015)	0.522	0.593	0.609	0.618	0.646
	(Dong and Zhang, 2016)	-	0.628	0.656	0.659	0.662
	HISK and ν-SVR	**0.703**	**0.716**	0.724	0.742	0.751
	BOSWE and ν-SVR	0.615	0.640	0.716	0.728	0.727
	HISK+BOSWE and ν-SVR	0.701	0.713	**0.737**	**0.754**	**0.779**
5→6	(Phandi et al., 2015)	0.187	0.539	0.662	0.680	0.713
	(Dong and Zhang, 2016)	-	0.647	0.700	0.714	0.750
	HISK and ν-SVR	0.715	0.726	0.754	0.757	0.781
	BOSWE and ν-SVR	0.617	0.623	0.644	0.650	0.692
	HISK+BOSWE and ν-SVR	**0.728**	**0.734**	**0.764**	**0.771**	**0.788**
7→8	(Phandi et al., 2015)	0.171	0.586	0.607	0.613	0.621
	(Dong and Zhang, 2016)	-	0.570	0.590	0.568	0.587
	HISK and ν-SVR	0.486	0.604	0.617	0.626	0.639
	BOSWE and ν-SVR	0.419	0.526	0.577	0.582	0.591
	HISK+BOSWE and ν-SVR	**0.522**	**0.606**	**0.637**	**0.638**	**0.649**

Table 3: Corss-domain automatic essay scoring results of our approach versus two state-of-the-art methods (Phandi et al., 2015; Dong and Zhang, 2016). Results are reported in terms of the quadratic weighted kappa (QWK) measure, using the same evaluation procedure as (Phandi et al., 2015; Dong and Zhang, 2016). The best QWK scores for each source→target domain pair are highlighted in bold.

nately, this is not possible because our approach works in the dual space and we cannot transform the dual weights into primal weights, as long as the histogram intersection kernel does not have an explicit embedding map associated to it. In future work, however, we aim to replace the histogram intersection kernel with the presence bits kernel, which will enable us to perform an error analysis based on the overused or underused patterns, as described by Ionescu et al. (2016).

4 Conclusion

In this paper, we described an approach based on combining string kernels and word embeddings for automatic essay scoring. We compared our approach on the Automated Student Assessment Prize data set, in both in-domain and cross-domain settings, with several state-of-the-art approaches (Phandi et al., 2015; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018). Overall, the in-domain and the cross-domain comparative studies indicate that string kernels, both alone and in combination with word embeddings, attain the best

performance on the automatic essay scoring task. Using a shallow approach, we report better results compared to recent deep learning approaches (Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018).

Acknowledgments

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of ACL*. pages 715–725.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v. 2.0. *Journal of Technology, Learning, and Assessment* 4(3):1–30.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Lan-

guage Model. *Journal of Machine Learning Research* 3:1137–1155.

Andrei Butnaru and Radu Tudor Ionescu. 2017. From Image to Text Classification: A Novel Approach based on Clustering Word Embeddings. In *Proceedings of KES*. page 17841793.

Chih-Chung Chang and Chih-Jen Lin. 2011. LibSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of EMNLP*. pages 1741–1752.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of ICML*. pages 160–167.

Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cdric Bray. 2004. Visual categorization with bags of keypoints. *In Workshop on Statistical Learning in Computer Vision, ECCV* pages 1–22.

Semire Dikli. 2006. An Overview of Automated Scoring of Essays. *Journal of Technology, Learning, and Assessment* 5(1):1–35.

Fei Dong and Yue Zhang. 2016. Automatic Features for Essay Scoring – An Empirical Study. In *Proceedings of EMNLP*. pages 1072–1077.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In *Proceedings of CONLL*. pages 153–162.

Peter W. Foltz, Darrell Laham, and Thomas K Landauer. 1999. Automated essay scoring: Applications to educational technology. In *Proceedings of EdMedia*. pages 40–64.

Rosa M. Giménez-Pérez, Marc Franco-Salvador, and Paolo Rosso. 2017. Single and Cross-domain Polarity Classification using String Kernels. In *Proceedings of EACL*. pages 558–563.

Radu Tudor Ionescu. 2015. A Fast Algorithm for Local Rank Distance: Application to Arabic Native Language Identification. In *Proceedings of ICONIP*. volume 9490, pages 390–400.

Radu Tudor Ionescu and Andrei Butnaru. 2017. Learning to Identify Arabic and German Dialects using Multiple Kernels. In *Proceedings of VarDial Workshop of EACL*. pages 200–209.

Radu Tudor Ionescu and Marius Popescu. 2016. UnibucKernel: An Approach for Arabic Dialect Identification based on Multiple String Kernels. In *Proceedings of VarDial Workshop of COLING*. pages 135–144.

Radu Tudor Ionescu and Marius Popescu. 2017. Can string kernels pass the test of time in native language identification? In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. pages 224–234.

Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2014. Can characters reveal your native language? a language-independent approach to native language identification. In *Proceedings of EMNLP*. pages 1363–1373.

Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2016. String kernels for native language identification: Insights from behind the curtains. *Computational Linguistics* 42(3):491–525.

Leah S. Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of SIGIR*. pages 90–95.

Thomas Leung and Jitendra Malik. 2001. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *International Journal of Computer Vision* 43(1):29–44.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Christopher J. C. H. Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research* 2:419–444.

David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2):91–110.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*. pages 3111–3119.

Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 2015. Flexible Domain Adaptation for Automated Essay Scoring Using Correlated Linear Regression. In *Proceedings of EMNLP*. pages 431–439.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of CVPR*. pages 1–8.

Marius Popescu and Cristian Grozea. 2012. Kernel methods and string kernels for authorship analysis. In *Proceedings of CLEF (Online Working Notes/Labs/Workshop)*.

Marius Popescu, Cristian Grozea, and Radu Tudor Ionescu. 2017. HASKER: An efficient algorithm for string kernels. Application to polarity classification in various languages. In *Proceedings of KES*. pages 1755–1763.

Marius Popescu and Radu Tudor Ionescu. 2013. The Story of the Characters, the DNA and the Native Language. *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications* pages 270–278.

508

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical Chaining for Measuring Discourse Coherence Quality in Test-taker Essays. In *Proceedings of COLING*. pages 950–961.

Wei Song, Dong Wang, Ruiji Fu, Lizhen Liu, Ting Liu, and Guoping Hu. 2017. Discourse Mode Identification in Essays. In *Proceedings of ACL*. pages 112–122.

J. A. K. Suykens and J. Vandewalle. 1999. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters* 9(3):293–300.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of EMNLP*. pages 1882–1891.

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2018. SkipFlow: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring. In *Proceedings of AAAI*. pages 1–8.

Andrea Vedaldi and B. Fulkerson. 2008. VLFeat: An Open and Portable Library of Computer Vision Algorithms. http://www.vlfeat.org/.

Jinhao Wang and Michelle Stallone Brown. 2008. Automated essay scoring versus human scoring: A correlational study. *Contemporary Issues in Technology and Teacher Education* 8(4):310–325.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2014. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of ACL*. pages 180–189.