Linguistic Driven Feature Selection for Text Classification as Stop Word Replacement

Daniel Schönle [1,*], Christoph Reich [1], and Djaffar Ould Abdeslam [2]

[1] Das Institut für Data Science, Cloud Computing und IT-Sicherheit (IDACUS), Furtwangen University, Furtwangen, Germany; Email: rch@hs-furtwangen.de (C.R.)

[2] L'Institut de Recherche en Informatique, Mathématiques, Automatique et Signal (IRIMAS), Université de Haute Alsace, Mulhouse, France; Email: djafar.ould-abdeslam@uha.fr (D.O.A.)

*Correspondence: daniel.schoenle@hs-furtwangen.de (D.S.)

***Abstract*—The common corpus optimization method "stop words removal" is based on the assumption that text tokens with high occurrence frequency can be removed without affecting classification performance. Linguistic information regarding sentence structure is ignored as well as preferences of the classification technology. We propose the Weighted Unimportant Part-of-Speech Model (WUP-Model) for token removal in the pre-processing of text corpora. The weighted relevance of a token is determined using classification relevance and classification performance impact. The WUP-Model uses linguistic information (part of speech) as grouping criteria. Analogous to stop word removal, we provide a set of irrelevant part of speech (WUP-Instance) for word removal. In a proof-of-concept we created WUP-Instances for several classification algorithms. The evaluation showed significant advantages compared to classic stop word removal. The tree-based classifier increased runtime by 65% and 25% in performance. The performance of the other classifiers decreased between 0.2% and 2.4%, their runtime improved between −4.4% and −24.7%. These results prove beneficial effects of the proposed WUP-Model.**

***Keywords*—text classification, natural language processing, feature selection, linguistics**

I. Introduction

The standard procedure "Stop Words Removal" can be used on Text Corpora to enhance Text Classification. It assumes that text tokens (filter criteria) with high occurrence frequency can be removed without affecting the classification (relevance criteria). Publicly available stop word lists have been published in the past based on language-specific analyses. These words are removed from the text corpus. The intended benefits are data size reductions, runtime reductions or classification performance improvements.

Word lists have drawbacks in terms of flexibility. Several methods in text pre-processing alter the words, e.g., Lemmatization converts the words into infinitives. There are studies proposing an adaption of stop words to document context [1]. Stop word lists omit information on the part of speech but this syntax information is part of the information value of a word. Basic linguistics suggests that classification relevance might correlate with Part-of-Speech (POS) [2]. We assume POS to be an important piece of information for text classification. In our approach, syntax is extracted by linguistic tokenizers. This information is retained by combining token and POS information. We propose classification importance as relevance value and word type as a filtering criterion. As a result, the proposed removal lists contain only POS.

Another aspect of stop word lists is their focus on language statistics. The classification method is not taken into account. We assume that classification methods have distinguishable relevance in terms of the POS.

We develop a definition of the importance of POS. The Part of Speed Weighted Importance (PWI) is based on relevance indicators in the classification process and a weighting of the degree of success of the classification result. Based on PWI, we present a procedure to generate a list of the least important POS of a classifier. The procedure is represented by the Weighted Unimportant Part of Speech Model (WUP-Model), the enactment of the model is represented by model instantiation (WUP-Instance). The WUP-Instance contains a list of the least important part of speech for the specific classifier.

We conduct a proof-of-concept and a comparative evaluation. We use chunks of datasets as document representation. To minimize variances and side effects, we use the TF-IDF method for vectorization and statistical methods for classification. Metrics for evaluation are balanced accuracy and runtime. The classic stop-word removal is used as baseline.

The effects are similar for almost all classification methods. For every Classifier a WUP-Instance has been successfully instantiated. All WUP-Instances differ, all classifiers have different preferences. The effects of the removal process are impressive. The tree-based classifier increased runtime by 65% and increased 25% in performance. The performance of the other classifiers decreased between 0.23% and 2.42%, their runtime improved between −4.4% and −24.7%. This result demonstrates the advantages of the PWI and the WUP-Model [3–5].

The datasets are listed in Table I.

Manuscript received October 8, 2022; revised December 27, 2022; accepted January 30, 2023; published August 11, 2023.

doi: 10.12720/jait.14.4.796-802

TABLE I. DATASETS

Dataset	Subset	Public	Docs	Chunks
Acorns [6]	roseslr	No	325	1
Acorns [6]	rosesva	No	325	1
AG News [7]	default	Yes	120,000	160
DBpedia14 [8]	dbpedia14	Yes	560,000	2100
Financial P.B. [9]	s.75agree	Yes	2417	0
GoEmotions [10]	admiration	Yes	142,858	10
GoEmotions [10]	approval	Yes	142,858	10
GoEmotions [10]	neutral	Yes	142,858	10
hatespeech18 [11]	default	Yes	7661	12
IMDb [12]	plaintext	Yes	25,000	20
Poem S. [13]	default	Yes	892	4
SMS Spam [14]	plaintext	Yes	7805	3
SST [15]	default	Yes	8544	4
TREC [16]	default	Yes	5452	8
TweetEval [17, 18]	emotion	Yes	3257	8
TweetEval [19]	hate	Yes	9000	4
TweetEval [20]	irony	Yes	11,917	2
TweetEval [21]	offensive	Yes	2863	1
TweetEval [22]	sentiment	Yes	45,615	48
TweetEval [23]	s.climate	Yes	355	3
TweetEval [23]	s.hillary	Yes	620	3
Web of Science [24]	WOS11967	Yes	16,754	66
Web of Science [24]	WOS46985	Yes	65,779	220
Web of Science [24]	WOS5736	Yes	8030	44
Total			1,331,185	2742

A. *Background*

Classifying text involves several processing steps in a pipeline, which differ depending on the classification technology used. In the pre-processing step, methods are available to divide the text into tokens, to simplify the words or to remove irrelevant tokens [25]. For the step of projection, the text into the feature space, static methods such as TDF-IDF or word embeddings such as word2vec can be used. The technologies used for the classification step can be divided into two categories. The Shallow Learning Classification Methods include Support Vector Machines (SVM), Random Forests (RF), Stochastic Gradient Descent (DGD), Naïve Bayes (NB), among others.

In the pre-processing method of tokenization, the document is converted into a collection or sequence of tokens. These tokens are used by the classifier to relate the document to the classes. For the classifier tokens have different degrees of relevance or importance. The information about this importance can be calculated for classification technologies that use linear methods. For other methods, the effort is higher or even cannot be extracted.

B. *Overview*

This research contributes a pre-processing method which reduces the size of data corpus without affecting text classification performance. The results are a higher effort in pre-processing, but reduced data sizes and shorter computation times in the modelling and classification phase.

The subsequent chapters are structured as follows: Section II contains related work focusing on the use of syntax information and feature selection. Section III defines the weighted importance of part of speech, based on two research hypotheses. The concept is presented in layers: (1) At the model level, the WUP-Model defines the requirements and procedures. (2) The creation of a WUP-Instance is based on the selection of classifier technologies and the data at instance layer. (3) The modification of the data by applying the WUP-Instance is described at the enactment level. Section IV presents and discusses the results. The conclusion and an outlook are given in Section V.

II. RELATED WORK

Syntax information has been used for Automatic Term Extraction [26], Text Ranking [27] and special language models for Pakistani language [28], and for Arabic language [29].

Syntax information for specialized text classifications includes sentiment classification [30], fake news detection [31], and hate speech detection [32]. In language specific research it is used for Gurmukhi language [33] and for Hindi language [34].

Research on stop word lists began in the 1980s [35]. Automatic extraction of Stop Words was based on token frequency. Extracting linguistic information had in studies of the 1990s a negative balance of computational cost [36, 37]. A simplified setup including a reduced set of word types showed minor effects on Rocchio and Support Vector Machines methods [5].

The focus on accelerating classification by reducing data without affecting results was studied by Alshanik and Apon *et al.* [1]. The focus was on an optimization for specific domain by analysing the token vector space. Information gain by using linguistic information studied regarding to benefits of learning languages [38, 39]. Feature selection for based on importance for text classification has been studied, resulting in the Gini coefficient of inequality [40], a Weighted Naïve Bayes [41]. The approach by Qin and Song *et al.* [42] uses part of speech for Chinese language.

III. WEIGHTED IMPORTANCE OF PART OF SPEECH

The concept consists of a definition of token importance and a three-layer procedure: at layer 1 definition of requirements and procedures (WUP-Model), at layer 2 instantiation on classifier and dataset (WUP-Instance), and at layer 3 filtering of data corpus (WUP-Enactment).

The overall complexity is minimalistic due to the experimental nature of this approach. In decisions simplicity and explainability was preferred to achieve generalization. An efficient proof-of-concept is used for instantiation and enactment. Objectives for each layer are distilled.

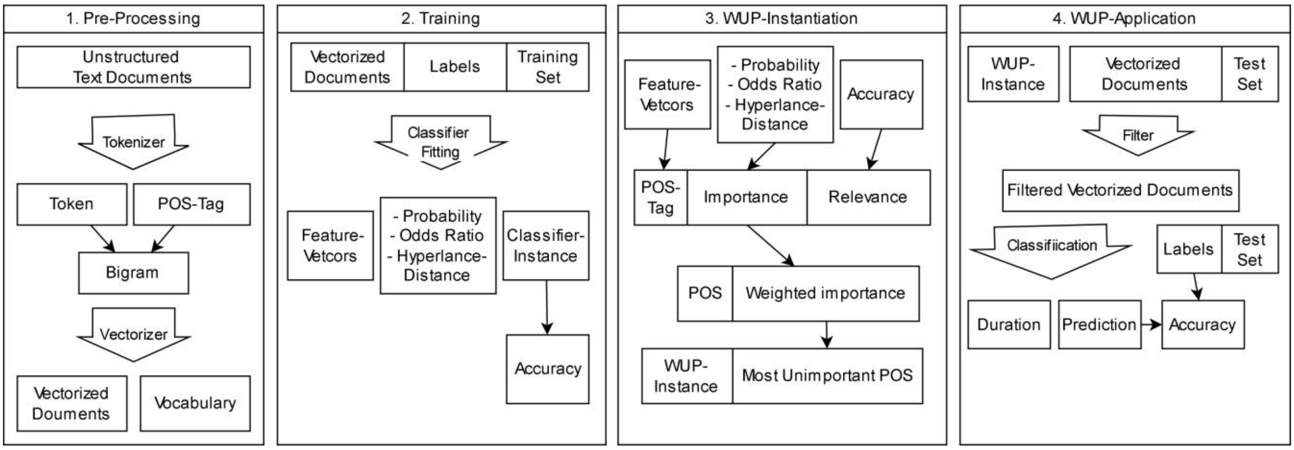

Figure 1. Pipeline-Integration of WUP-Concept.

A. *Research Hypotheses*

The presented concept for sophisticated stop words uses linguistic expert knowledge and insight of machine learning techniques. The resulting stop word list contains part of speech tags instead of words and depends on the used classifier technique and hyperparameter configuration.

We present a concept for using linguistic knowledge to support binary text classification. The foundation is formed by three hypotheses:

(1) There are tokens of the same part of speech whose classification relevance is similar.

(2) The classification relevance of part of speech varies depending on the classification method.

Automatic text classification benefits from the removal of tokens whose part of speech is of low relevance to the classification.

B. *WUP-Model*

Requirements and procedures have to be defined, representing the WUP-Model. An overview of WUP-Model integration is shown in Fig. 1.

Validation Requirements: The WUP-Model involves the use of high-dimensional data (features) and predictive statistical models (classifiers). Statistical issues and potential confounding variables have to be avoided to validate the WUP-Model [43, 44]. This requires a high variance of data and classification problems as well as a selection of methods and models with minor impact factors, low complexity, and high tractability. The text documents should feature variance in text length (e.g., word count) and text type (e.g., sentence, short text, text). The classification task should vary (e.g., topic detection, sentiment detection).

Tagging and Vectorization: Linguistic information of part of speech of the documents is obtained by a part of speech tagger. The tagger creates tokens and syntax information. The selected SpacyTagger uses the glossary of the Penn Treebank Project [45, 46]. The syntax information is adjunct to the tokens by building bigrams. As the frequency of parts of speech is unbalanced, dependency information is used for nouns and verbs in addition to part of speech. This yields to an impact in the vectorization process. A context-based word encoding, such as Word2Vwc or GloVE rely on general word definitions. Such a word encoding method has to be an adapted for using for bi-grams, i.e., to ignore the syntax part of the bi-gram. An adaption for content-based vectorization is not necessary. The most developed content-based vectorization technology, term frequency-inverse document frequency (TF-IDF) calculates word vectors and provides a vocabulary, documenting vector-feature assignments. The size limit of the vocabulary is removed to keep all features. The number of distinguishable features is increased by building bigrams (tokens to the power of part speech tags). Lemmatization was applied as countermeasure.

Weighted Importance: The classifier-model is fitted on the vectorized documents, handling the bi-gram-vectors as features. The Classifier-Algorithm must provide the classification importance of features. These features are transformed into bigrams using the vectorizer-vocabulary and part of speech information is extracted for every part of speech tag. The classification-performance for every dataset-chunk relative to all chunks is calculated. The importance value of the part of speech tags is weighted by the relative success rate leading to the ‘part of speech weighted relevance’ (PWR). All PWR-Values of part of speech must be below 2% of all PWR-Values. As part of speech may have several PWR-Values, appropriate filtering is applied. Instance-Level

At instance-level the model is used to extract the most unimportant tokens (WUP-Instance) in a proof-of-concept. For every classifier configuration WUP-Instance is different as the importance depends on classifier technology and classifier configuration. The set of important and unimportant tokens are subsets of the set of tokens of the dataset. The WUP-Instance correlates with the dataset. The generalization of the WUP-Instance increases with the dataset size.

Classifier Selection: Machine learning based document classification has benefited greatly with deep- learning-based technologies such as transformers. However, these technologies have drawbacks. Transformers technologies have weaknesses in the explainability of the results [47], multilingualism [48], homogenization [49], data bias [50], and overall understanding the technology [51].

Shallow classifier technologies like random forest or support vector machines have still advantages. In specific settings, like low number of samples, the performance of deep-learning approaches is comparable to shallow approaches [52].

Bias in the instance-level and enactment-level tamper the results of the model evaluation. Auditability, repeatability and explainability reduce or at least allow at detection of bias occurred by classifiers. In addition, explainability is essential to extract the importance of tokens. The de-facto-performance of the classification technology is irrelevant, as stated earlier.

Classifiers that provide auditability, repeatability and explainability [53] were selected according [25] as listed in Table II. This includes Naïve Bayes [54, 55], Random Forest [56], Logistic Regression [57] and, Support Vector Machine [58].

TABLE II. WUP-INSTANCES IN POC

Classifier	WUP-Instance *	
	Size	Tags
SVM**	222	SYM, JJ-punct, JJ-relcl, RP, NNP-aux, CC, NNauxpass, NN-nummod, JJ-meta, AFX, NNP, XX NNS-mark, NN-acl, NNS-auxpass, NN-aux, NNSrelcl, NNP-agent, ...
RF**	91	NNP-advcl, JJ-advmod, NNP-nmod, JJ-acomp NNP-nsubjpass, NNP-attr, NNP-amod, JJ-advcl NNS-conj, NN-nsubjpass, WP, NN-oprd, NNPposs, NN-poss, NNS-appos, RBS, ...
NB	10	-LRB-, LS, ADD, TO, FW, WP$, SYM, CD, EX XX
SGD	8	-LRB-, LS, AFX, FW, NFP, -RRB-, SYM, CD
GridSVM	2	SYM, CD

*Format of Elements: Part-of-Speech-Tag[-Dependency-Tag]; Part-of-Speech-Tag and Dependency-Tag according Penn Treebank [45, 46].
**Incomplete Listing.

Classifiers have configuration parameters that influence their classification mechanism. The classifier-setups were developed by evaluating five setups for every classifier for the SST Dataset. For Naïve Bayes and Support Vector Machine different and kernels were selected, other hyperparameters were chosen by random. A setup using grid search optimised support vector machine was used as real-world example. The hyperparameters are optimized for every dataset-chunk. This setup changes the hyperparameters and may not deliver constant token importance. The instances are defined by their classifier technology, configuration and hyperparameter, as listed in Table II.

Building the data ensemble: The data basis consists of an ensemble of datasets. The data sets are either public or can be requested from the cited sources, see Table I for details. Usual datasets intend are performance evaluations of classification models for a specific task, i.e., hate speech detection or short text classification for a specific kind of document style. The WUP-Model is classification task and document style independent. The weight relevance of tokens relies on classification performance, the importance on classification model. The data was selected to achieve a high degree of variance in tasks and document attributes. The datasets have been split up into chunks of sufficient size to reduce impact of size variance and increase granularity. The training/test sets given by the dataset were combined and recreated with adapted test set sizes.

The result of the instantiating is listed in Table III. For every classifier setup, a list of part of speech has been successfully created. Theses removal lists can be applied per classifier setup on every text in English.

TABLE III. CLASSIFIERS IN POC

Classifier	Relevant Hyper Parameters*	Relevant Attributes*	Config**
SVM	Regularization	Effective in high dimensional spaces and robust against overfitting	alpha = 1.0, linear Kernel
RF	Depth, Estimators, Features	Prone to overfitting	max_depth=5, n_estimators=400, max_features=10
NB	Model	Strong Assumptions	Bernoulli Model
SGD	Loss function, Penalty, Regularization multiplicator, Iterations	Strong Assumptions	oss=modified_huber, penalty=l2, alpha=1e-3, max_iter=14
Gridsearch	Parameter Space, Score Function	Adaption of Hyperparameters	C: 1, 10, 100, 1000, gamma: 1, 0.1, 0.01, 0.001, 0.0001, score=accuracy, Linear Kernel SVM

*According [25]. **Incomplete Listing.

D. Enactment-Level

The process of applying an instance on a dataset is similar to apply stop words. The instances are used on data after applying a part of speech tagger as it relies on part of speech tags. The enactment can take part in the pre-process phase or classification phase. The later one was chosen, as the datasets were already pre-processed, and the WUP-Model is classifier-setup specific. To gain further insights the removal according the WUP-Model was applied per sentence.

E. Proof-of-Concept

The objectives are as follows:

(1) Linguistic Information Injection. Develop a method to inject part of speech information into documents. As the importance of parts of speech

is required, the syntax information has to be available to the classifier.

(2) Establish a Dataset-Ensemble. A Dataset-Ensemble consisting of different text types, document types, language types and classification problems hast to be established.

(3) Select classifiers. Text Classification Technology based on linear methods has to be to be selected. Configurations has to be defined.

(4) Unimportant Part of Speech Model. Implement a model for weighted unimportant part of speech (WUP-Model). Based on the importance of tokens, a procedure to instantiate the model that provides a list of unimportant tokens has to be developed.

(5) Model Instance. Instantiate the WUP-Model. The WUP-Model has to be applied on an appropriate set of classifiers.

(6) Instance Enactment. Apply WUP-Instance. The WUP-Model should be applicable on datasets either in the pre-processing steps of documents or in the classification step.

The proposed proof-of-concept is designed as pilot study. Compromises were made in favour of simplicity at the expense of performance and expressiveness. The proof-of-concept serves exclusively to test the hypotheses. The absolute classification performance is not considered relevant. The chosen shallow classifiers are outperformed by state-of-the-art classifiers for overall text classification. Only one classifier-setup per classifier-technology has been used. The vectorization by TF-IDF is outdated and the dictionary size has not been optimized. The balance of the datasets was not handled. The effects of unbalanced classification may lead to a deterioration in performance.

IV. Results and Discussion

The objectives of the evaluation of the concept are:

(1) Evaluate WUP-Model by instantiation analysis.

(2) Evaluate the effects of the enactment of WUP-Instances.

A. Instances

The WUP-Instances of the POC are listed in Table III and Table IV. For every classifier-setup an instance could be established. The part of speech symbol and cardinal number is included in every instance.

The importance of tokens for classification can be distinguished by syntax information, the WUP-Model incorporates part of speech and syntax dependencies. The information gain on tokens grouped by linguistic information (part of speech and dependency) differs for all classifier technologies. The instances per classifier have high variance. The Gradient Descent and Naive Bayes classifier have the highest similarity. This may occur as Gradient Descent and Naive Bayes' tendency to make strong assumptions, see Table II. Although GridSearchSVM adjusts the hyperparameters for each data chunk, a small instance could be created.

TABLE IV. WUP-Instance Excerpt

Classifier	Description of Tags
SGD*	left round bracket, list item marker, affix, foreign word, superfluous punctuation, right round bracket, symbol, cardinal number

*WUP-Instance contains Part-of-Speech-Tags only.

B. Enactment Results

To evaluate the effects of the WUP-Model the effects were measured against a baseline. Two metrics were chosen, balanced accuracy and runtime. As Baseline the stop-words-removal-approach is used. The results are listed in Table V.

The impact on Random Forest Classifier is high. Accuracy is increased by 25.24%. Since the enactment of the WUP-Model reduces the amount of data, the increase in runtime is surprising. The accuracy of all other classifiers remained stable under minor changes. Gradient Descent had significant runtime reductions, and GridSearchSVM also needed less time. The Support Vector Machine was 5.58% slower. This can be explained by a different source of importance. For Support Vector Machine this is the distance of the feature from the hyperplane.

TABLE V. WUP-Model Effects*

Classifier	Balanced Accuracy %	Duration %
Random Forest	25.24	65.63
Gridsearch SVM	−0.29	−11.83
Support Vector Machine	0.28	−5.58
Stochastic Gradient Descent	−0.47	−24.68
Naïve Bayes	−2.43	−4.42

*Compared to standard stop word list effects.

V. Conclusion and Future Work

For every classifier-setup an instance could be established. Every Instance has a different size and content. Research Hypotheses (a) and (b) are validated. With the exception of the Support Vector Machine, all classifiers were able to significantly reduce the runtime (Table V). Research Hypotheses (c) is validated. It has been shown that the classifiers have different preferences for part of speech, but the robustness and generalisation of the proof-of-concept is limited. A detailed model in which the concept is implemented in more detail should be created.

A major future objective is the validation of the WUP-Model for a whole language. Derived future objectives are studies on effects by the statistical methods, statistics per token, an increased database, and the integration of state of the technologies like word models.

Overall, the integration of the WUP-Model in the pre-processing of text classification is desirable.

Conflict of Interest

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

The research is conducted under the supervision of Christoph Reich and Djaffar Ould Abdeslam. The initial and the final version was written by Daniel Schönle. Daniel Schönle conducted the research, created and evaluated the algorithm, and wrote the paper. Christoph Reich and Djaffar Ould Abdeslam reviewed the manuscript and supervised the research. All authors had approved the final version of the paper.

FUNDING

This Research was funded by the Federal Ministry of Education and Research of Germany in framework of FiSK (Project-Number 16DHB4005).

REFERENCES

[1] F. Alshanik, A. Apon, A. Herzog, I. Safro, and J. Sybrandt, "Accelerating text mining using domain-specific stop word lists," in *Proc. 2020 IEEE International Conference on Big Data*, IEEE, 2020, pp. 2639–2648.

[2] D. Bouchard, *The Semantics of Syntax: A Minimalist Approach to Grammar*, University of Chicago Press, 1995.

[3] R. W. Brown, "Linguistic determinism and the part of speech," *The Journal of Abnormal and Social Psychology*, vol. 55, no. 1, p. 1, 1957.

[4] A. E. Goldberg, *Constructions: A Construction Grammar Approach to Argument Structure*, John Benjamins Publishing Company, 1995.

[5] A. Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study," in *Proc. European Conference on Information Retrieval*, 2004, pp. 181–196.

[6] R. H. Nehm, E. P. Beggrow, J. E. Opfer, and M. Ha, "Reasoning about natural selection: Diagnosing contextual competency using the acorns instrument," *The American Biology Teacher*, vol. 74, no. 2, pp. 92–98, 2012.

[7] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. NIPS'15: the 28th International Conference on Neural Information Processing Systems*, 2015, pp. 649–657.

[8] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. V. Kleef, S. Auer, *et al*., "Dbpedia—A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[9] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *Journal of the Association for Information Science and Technology*, vol. 65, issue 4, pp. 782–796, 2014.

[10] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A dataset of fine-grained emotions," in *Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[11] O. Gibert, N. Perez, A. García-Pablos, and M. Cuadros, "Hate speech dataset from a white supremacy forum, in *Proc. the 2nd Workshop on Abusive Language Online (ALW2)*, October 2018, pp. 11–20.

[12] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, June 2011, pp. 142–150.

[13] E. Sheng and D. Uthus, "Investigating societal biases in a poetry composition system," arXiv pre-print, arXiv:2011.02686, 2020. https://dl.acm.org/doi/abs/10.1145/2034691.2034742

[14] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: New collection and results," in *Proc. the 2011 ACM Symposium on Document Engineering (DOCENG'11)*, 2011.

[15] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. the 2013 Conference on Empirical Methods in Natural Language Processing*, October 2013, pp. 1631–1642.

[16] X. Li and D. Roth, "Learning question classifiers," in *Proc. COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[17] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko. "Semeval-2018 task 1: Affect in tweets," in *Proc. the 12th International Workshop on Semantic Evaluation*, 2018, pp. 1–17.

[18] F. Barbieri, J. Camacho-Collados, L. Espinosa-Anke, and L. Neves, "TweetEval: Unified benchmark and comparative evaluation for tweet classification, in *Proc. Findings of EMNLP*, 2020.

[19] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, "SemEval2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter," in *Proc. the 13th International Workshop on Semantic Evaluation*, 2019, pp. 54–63.

[20] C. V. Hee, E. Lefever, and V. Hoste, "Semeval-2018 task 3: Irony detection in English tweets," in *Proc. the 12th International Workshop on Semantic Evaluation*, 2018, pp. 39–50.

[21] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)," in *Proc. the 13th International Workshop on Semantic Evaluation*, 2019, pp. 75–86.

[22] S. Rosenthal, N. Farra, and P. Nakov, "Semeval-2017 task 4: Sentiment analysis in twitter," in *Proc. the 11th international workshop on semantic evaluation (SemEval-2017)*, 2017, pp. 502–518.

[23] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proc. the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 31–41.

[24] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *Proc. the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2017.

[25] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification algorithms: From text to predictions," *Information*, vol. 13, no. 2, 83, 2022.

[26] N. I. Simon and V. Kešelj, "Automatic term extraction in technical domain using part-of-speech and common-word features," in *Proc. the ACM Symposium on Document Engineering*, 2018, pp. 1–4.

[27] J. Lin, R. Nogueira, and A. Yates, "Pretrained transformers for text ranking: Bert and beyond," *Synthesis Lectures on Human Language Technologies*, vol. 14, no. 4, pp. 1–325, 2021.

[28] M. Yousaf, A. Habib, I. A. Khan, and F. Masroor, "A Corpus based study language fixity in journalistic discourse: A Corpus-based study of Pakistani editorials and blogs," *Corporum: Journal of Corpus Linguistics*, vol. 2, no. 2, pp. 55–67, 2019.

[29] B. Abu-Salih, "Applying vector space model (VSM) techniques in information retrieval for Arabic language," arXiv pre-print arXiv:1801.03627, 2018.

[30] E. S. Usop, R. R. Isnanto, and R. Kusumaningrum, "Part of speech features for sentiment classification based on latent Dirichlet allocation," in *Proc. 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2017, pp. 31–34.

[31] M. Mahyoob, J. Al-Garaady, and M. Alrahaili, "Linguistic-based detection of fake news in social media," *International Journal of English Linguistics*, vol. 11, no. 1, 2020.

[32] S. Bhatt, N. Goenka, S. Kalra, and Y. Sharma, "Fake news detection: Experiments and approaches beyond linguistic features," in *Data Management, Analytics and Innovation*, Springer, 2022, pp. 113–128.

[33] K. Jasleen and R. S. Jatinderkumar, "POS word class based categorization of Gurmukhi language stemmed stop words," in *Proc. the First International Conference on Information and Communication Technology for Intelligent Systems*, 2016, vol. 2, pp. 3–10.

[34] R. Rani and D. K. Lobiyal, "Automatic construction of generic stop words list for Hindi text," *Procedia Computer Science*, vol. 132, pp. 362–370, 2018.

[35] W. J. Wilbur and K. Sirotkin, "The automatic identification of stop words," *Journal of Information Science*, vol. 18, no. 1, pp. 45–55, 1992.

[36] T. Strzalkowski, J. P. Carballo, and M. Marinescu, "Natural language information retrieval: Trec-3 report," *NIST Special Publication SP*, p. 39, 1995.

[37] E. M. Voorhees and D. Harman, "Overview of the sixth text retrieval conference (TREC-6)," *Information Processing & Management*, vol. 36, no. 1, pp. 3–35, 2000.

[38] M. Subasini and B. Kokilavani, "Significance of grammar in technical English," *International Journal of English Literature and Culture*, vol. 1, no. 3, pp. 56–58, 2013.

[39] D. N. Jureddi and N. Brahmaiah, "Barriers to effective communication," *Journal of English Language and Literature*, vol. 3, no. 2, pp. 114–115, 2016.

[40] R. Sanasam, H. Murthy, and T. Gonsalves, "Feature selection for text classification based on Gini coefficient of inequality," in *Proc. PMLR: Feature Selection in Data Mining*, 2010, pp. 76–85.

[41] S. Ruan, B. Chen, K. Song, and H. Li, "Weighted naïve bayes text classification algorithm based on improved distance correlation coefficient," *Neural Computing and Applications*, vol. 34, no. 4, pp. 2729–2738, 2022.

[42] S. Qin, J. Song, P. Zhang, and Y. Tan, "Feature selection for text classification based on part of speech filter and synonym merge," in *Proc. 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, IEEE, 2015, pp. 681–685.

[43] A. E. Teschendorff, "Avoiding common pitfalls in machine learning OMIC data science," *Nature Materials*, vol. 18, no. 5, pp. 422–427, 2019.

[44] R. Dinga, L. Schmaal, B. W. J. H. Penninx, D. J. Veltman, and A. F. Marquand, "Controlling for effects of confounding variables on machine learning predictions," *BioRxiv*, 2020, https://doi.org/10.1101/2020.08.17.255034

[45] B. Santorini, "Part-of-speech tagging guidelines for the Penn treebank project," Technical Reports (CIS), Department of Computer & Information Science, University of Pennsylvania, 1990.

[46] Department of Linguistics. (2003). Alphabetical list of part-of-speech tags. [Online]. Available: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos. html

[47] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *Proc. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018, pp. 210–215.

[48] J. Singh, B. McCann, R. Socher, and C. Xiong, "Bert is not an interlingua and the bias of tokenization," in *Proc. the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, 2019, pp. 47–55.

[49] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al*., "On the opportunities and risks of foundation models," arXiv pre-print, arXiv:2108.07258, 2021. https://arxiv.org/abs/2108.07258

[50] A. V. González, M. Barrett, R. Hvingelby, K. Webster, and A. Søgaard, "Type b reflexivization as an unambiguous testbed for multilingual multi-task gender bias," arXiv pre-print, arXiv:2009.11982, 2020.

[51] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020.

[52] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From shallow to deep learning," arXiv pre-print, arXiv:2008.00364, 2020, https://arxiv.org/abs/2008.00364

[53] C. C. Aggarwal and C. X. Zhai, "A survey of text classification algorithms," *Mining Text Data*, pp. 163–222, 2012.

[54] S. Xu, Y. Li, and Z. Wang, "Bayesian multinomial naive bayes classifier to text classification," *Advanced Multimedia and Ubiquitous Engineering*, pp. 347–352, 2017.

[55] R. Krishnapuram and J.M. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, 1993.

[56] T. K. Ho, "Random decision forests," in *Proc. the 3rd International Conference on Document Analysis and Recognition*, 1995, vol. 1, pp. 278–282.

[57] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. the Fifth Annual Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[58] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.