JCT at SemEval-2022 Task 4-A: Patronism Detection in Posts Written in English using Pre-processing Methods and various Machine Learning Methods

Yaakov HaCohen-Kerner, Ilan Meyrowitsch, Matan Fchima

Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
21 Havaad Haleumi St., P.O.B. 16031, 9116001 Jerusalem, Israel
kerner@jct.ac.il, meyrowitsch@gmail.com, matanf1992@gmail.com

Abstract

This paper describes our submissions to SemEval-2022 subtask 4-A - “Patronizing and Condescending Language Detection: Binary Classification". We developed different models for this subtask. We applied 11 supervised machine learning methods and 9 pre-processing methods. Our best submission was a model we built with BertForSequenceClassification. Our experiments indicate that pre-processing stage is a must for a successful model. The dataset for Subtask 1 is highly imbalanced. The F1-scores on the oversampled imbalanced training dataset were higher than the results on the original training dataset.

1 Introduction

The explosion of social media in recent years also enables an increasing the number of patronizing and condescending language (PCL). Patronizing language is best described as expressions that are agreeable and show kindness to a person or group in a condescending manner, indicating that the person or group is inferior (McCune and Matthews, 1978).

The discourse of condescension has three main characteristics: (1) It does not contain anything openly critical or negative, and often contains insincere praise; (2) it assumes a difference in status and worth between the writer and the person who wrote about him; and (3) this assumed difference is disputed by the listener (Huckin, 2002).

PCL can harm individuals or groups of people and may cause harmful effects on society. Therefore, it is important to develop efficient computerized systems capable of detecting PCL (Lo and Wei, 2006).

PCL detection is not a simple problem because it requires understanding the context of the situation, the relevant culture, and indirect clues. In social media texts, the problem is harder due to the different levels of ambiguities in natural language and the noisy nature of such texts.

In contrast to the offensive language or hate speech detection field, where there has been relatively an extensive research (e.g., Basile et al., 2019; Zampieri et al., 2019; Zampieri et al., 2020), PCL is still a relatively new and open field of study in Natural language processing (NLP) and machine learning (ML) (Pérez-Almendros et al., 2020).

Pérez-Almendros et al. (2020) introduced the Don’t Patronize Me! Dataset. This dataset contains paragraphs extracted from news stories, which have been annotated to indicate the presence of PCL at the text span level.

This paper describes our research and participation in subtask 4-A for patronism detection in posts written in English. The full description of task 4 in general and 4-A, in particular, is given in Perez-Almendros et al. (2022).

The structure of the rest of the paper is as follows. Section 2 introduces a background concerning patronism detection, text pre-processing, and TC with imbalanced classes. Section 3 describes subtask 4-A and its training dataset. In Section 4, we present the submitted models and their experimental results. Section 5 summarizes and suggests ideas for future research.

2 Related Work

Various NLP methods have been applied in the detection of several types of harmful language such as offensive language or hate speech detection (Basile et al., 2019; Zampieri et al., 2019; Zampieri et al., 2020). Previous NLP tasks have generally focused on explicit, aggressive, and flagrant phenomena such as fake news detection (Conroy et al., 2015).

During the last three years, several studies on PCL have appeared. Wang and Potts (2019)

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 519 - 524
July 14-15, 2022 ©2022 Association for Computational Linguistics

introduced the task of modeling condescension and developed an annotated dataset of social media messages. Sap et al. (2019) discussed various implications behind certain uses of language. Mendelsohn et al. (2020) analyzed, from a computational linguistics viewpoint, how language has dehumanized minorities in media news.

2.1 Text preprocessing

Text preprocessing is an important step of TC in general and in social text documents in particular. Classification of text dataset that has not been carefully cleaned or preprocessed might lead to misleading results.

HaCohen-Kerner et al. (2019) investigated the impact of all possible combinations of six preprocessing methods (spelling correction, HTML tag removal, converting uppercase letters into lowercase letters, punctuation mark removal, reduction of repeated characters, and stopword removal) on TC in three benchmark mental disorder datasets. In another study, HaCohen-Kerner et al. (2020) explored the influence of various combinations of the same six basic preprocessing methods on TC in four general benchmark text corpora using a bag-of-words representation. The general conclusion was that it is always advisable to perform an extensive and systematic variety of preprocessing methods because it contributes to improving TC accuracy.

2.2 Text classification with imbalanced classes

The problem with TC with imbalanced classes is that there are too few examples of the minority class to effectively learn a good predictive TC model. There are various methods to cope with this problem (e.g., Liu et al., 2004). The main idea is to change the dataset until a more balanced distribution is reached. Two well-known sampling methods that enable such a change are oversampling and undersampling (e.g., Yap et al., 2014). Random oversampling means randomly duplicating examples in the minority class. Random undersampling means randomly deleting examples in the majority class.

An additional frequent method is to generate synthetic samples, which means randomly sampling the attributes from instances in the minority class (Zhu et al., 2017). There are several algorithms that support the generation of synthetic samples. The most popular one is called the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla, 2002). This method is an oversampling method that creates synthetic samples from the minor class instead of creating copies. This method selects two or more similar instances and perturbs an instance one attribute at a time by a random amount within the difference to the similar instances.

Other possible methods are to try a variety of different types of machine learning (ML) methods in general and penalized variants of these methods that charge an additional cost on the model for making classification mistakes on the minority class during training.

Readers interested in expanding and deepening the topic of solutions to TC with imbalanced classes are referred to the following articles (Chawla et al., 2002; He and Ma, 2013; Krawczyk, 2016; Brownlee, 2020).

3 Task and Training Dataset Description

We only participated in subtask 4-A - “Patronizing and Condescending Language Detection: Binary Classification", which deals with the classification of each post as a patronizing or condescending language (PCL) or not in the English language. Table 1 presents various statistical details about the data set.

	not patronize	is patronize	total
Documents	9,476	993	10,469
% Docs	90.5	9.5	100
words	453,690	53,245	506,935
characters	2,514,890	286,435	2,801,325
avg word per doc	47.87	53.62	50.745
avg chars per doc	265.39	288.45	276.92
words std	32.77	28.62	30.695
chars std	158.36	175.52	166.94

Table 1: Details of the training set.

The analysis of the details presented in Table 1 shows that the dataset is highly imbalanced with a ratio of about 91:9 (not patronize: is patronize). We changed this rate to 77:23 by the creation of new partial 'patronized' posts extracted from various posts that belong to different categories of positive patronized labels available from TASK 6-2 (multi-label classification). We also evaluated an equal split (50:50) by duplication of the patronized sentences. However, the experimental results using the equal split lead to results that were lower than the results using unequal ratios. All the python code lines used for improving the ratio, preprocessing methods, and the different models are available on Github at https://github.com/meyrow/pcl-detection-task4-semeval2022.

4 The Submitted Models and Experimental Results

We applied 11 supervised ML methods to the training dataset. Seven of them were classical ML methods: Random Forest (RF), K Nearest Neighbours (KNN), Support Vector Classifier (SVC), XGBoost Classifier, Logistic Regression (LR), Decision Tree (DT), Naive Bayes (NB), and four of them were deep learning (DL) methods: Bert, DistilBert, Roberta, and Albert.

In our various models, we applied nine sub-types of preprocessing methods: remove newlines, remove HTML Tags, remove Links, remove White spaces, remove accented characters, conversion to lower-case, reduce repeated characters, and punctuations, expand contractions, and remove special characters.

These methods were applied using the following tools and information sources:

- The Python 3.7.3 programming language[1].
- Scikit-learn – a Python library for ML methods[2].
- Numpy – a Python library that provides fast algebraic calculous processing, especially for multidimensional objects[3].

In our experiments, we tried to find the best combination of ML method, preprocessing methods, and oversampling methods. The training set was split into 80:20 train: test and the training test was using 90 percent in every epoch to train and 10 percent of the training set was used for validation.

Figure 1 presents training and validation loss curves of our BERT model with 20 epochs showung that training and validation continuously improved themselves. We noticed that a gap between training and validation began to grow, therefore we had to stop the model after 20 epochs, otherwise, the model will be overfitted. Figures 2 and 3 present the confusion matrices of our BERT model with 20 epochs and the decision tree model, respectively. The confusion matrix of both models demonstrates that the dataset is imbalanced as shown in Table 1. We also noticed that the ratio of 77:23 after improving the original dataset is close to the ratio shown in the confusion matrix. That indicates that our models are well trained. To select the best model we compared the F1-score.

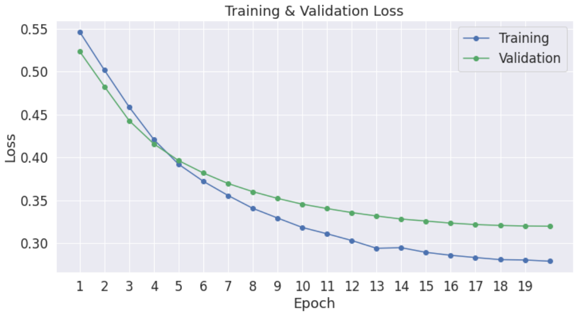

Figure 1: Training and validation loss curves of our BERT model with 20 epochs.

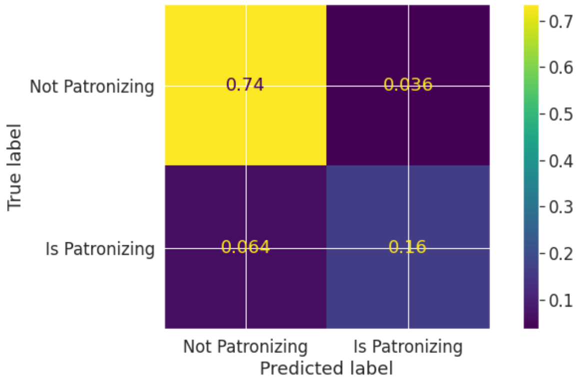

Figure 2: Confusion matrix of our BERT model with 20 epochs.

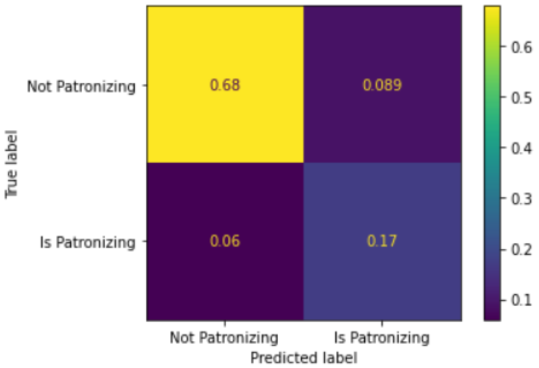

Figure 3: Confusion matrix of our Decision Tree model.

[1] https://www.python.org/downloads/release/python-373/

[2] https://scikit-learn.org/stable/index.html

[3] https://numpy.org

Our best submission was a model called Matanbert that we built using a function called BertForSequenceClassification. This BERT model includes 768 layers. Its values of the learning rate, epsilon, number of epochs, and batch size were 2e-7, 1e-8, 20, and 16, respectively. This model was ranked the 62nd position. Its F1-score over the PCL class, precision, and recall results are 0.377, 0.3536, and 0.4038, respectively.

Table 2 presents the results of the submitted models. The F1-score over the PCL class on the training dataset of our best model was 0. 77 while the F1-score over the PCL class on the test dataset of our best model was only 0.377. Currently, the posts' labels of the test dataset are unknown. Therefore, we do not have any definite explanation(s) for such a large decrease in the results. Possible explanations might be: (1) The training dataset is different in its balance rate than the balance rate of the competition test dataset and (2) the content of a relatively high number of news items in the competition test dataset is fundamentally different from the content of the news in the training dataset.

Our code is available on Github at https://github.com/meyrow/pcl-detection-task4-semeval2022. Our models are available for reproducibility with comments that explain the code and parameters such as epsilon, learning rate batch, and epochs.

5 Conclusions and Future Research

In this paper, we describe our submissions to subtask 4-A of the SemEval-2022 contest. We submitted the models that achieved the best results while trying to choose two models that applied different supervised learning methods.

Future research ideas include (1) Acronym disambiguation that will extend and enrich the social text and might enable better classification (e.g., HaCohen-Kerner et al., 2008; HaCohen-Kerner et al., 2010A); (2) use of skip character n- to overcome problems such as noise and sparse data (HaCohen-Kerner et al., 2017); (3) use of stylistic feature sets (HaCohen-Kerner et al., 2010B) and key phrases that can be extracted from text files (HaCohen-Kerner et al., 2007).

Model Name	split mode	ML Method	Features	F1-score over the PCL class on the training dataset	F1-score over the PCL class on the test dataset
Matan_bert	80:20	BertForSequence Classification	Layers: 768 Learning rate: 2e-7 Epsilon: 1e-8 Epochs: 20 Batch-size: 16	0.77	0.377
Matan_ Decision_Tree	70:30	Decision Tree	criterion= 'entropy' random_state = 0	0.7	was not published

Table 2: Results of the submitted models.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In Proceedings of SemEval.

Jason Brownlee. 2020. Imbalanced classification with python: Better metrics, balance skewed classes, cost-sensitive learning. Machine Learning Mastery.

Nitesh V. Chawla, Kevin W Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

Niall J. Conroy, Victoria L. Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology, 52(1):1–4.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Haibo He and Yunqian Ma (Eds.). 2013. Imbalanced learning: foundations, algorithms, and applications.

Thomas Huckin. 2002. Critical discourse analysis and the discourse of condescension. Discourse studies in composition, 155, 176.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. ACM Computing Surveys (CSUR), 50(5), 1-22.

Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz. 2008. Combined one sense disambiguation of abbreviations. In Proceedings of ACL-08: HLT, Short Papers, Association for Computational Linguistics, pages 61-64, Columbus, Ohio, Association for Computational Linguistics. URL: https://aclanthology.org/P08-2.

Yaakov HaCohen-Kerner, Ittay Stern, David Korkus, and Erick Fredj. 2007. Automatic machine learning of keyphrase extraction from short HTML documents written in Hebrew. Cybernetics and Systems: An International Journal, 38(1), 1-21.

Yaakov HaCohen-Kerner, Dror Mughaz, Hananya Beck, and Elchai Yehudai. 2008. Words as classifiers of documents according to their historical period and the ethnic origin of their authors. Cybernetics and Systems: An International Journal, 39(3), 213-228.

Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz. 2010A. HAADS: A Hebrew Aramaic abbreviation disambiguation system. Journal of the American Society for Information Science and Technology, 61(9), 1923-1932.

Yaakov HaCohen-Kerner, Hananya Beck, Elchai Yehudai, and Dror Mughaz. 2010B. Stylistic feature sets as classifiers of documents according to their historical period and ethnic origin. Applied Artificial Intelligence, 24(9), 847-862.

Yaakov HaCohen-Kerner, Ziv Ido, and Ronen Ya'akobov. 2017. Stance classification of tweets using skip char Ngrams. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 266-278). Springer, Cham.

Yaakov HaCohen-Kerner, Yair Yigal, and Daniel Miller. 2019. The impact of Preprocessing on Classification of Mental Disorders, in Proc. of the 19th Industrial Conference on Data Mining, (ICDM 2019), New York.

Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. 2020. The influence of preprocessing on text classification using a bag-of-words representation, PloS one, vol. 15, p. e0232525.

Bartosz Krawczyk. 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221-232.

Ven-hwei Lo, and Ran Wei. 2006. Perceptual differences in assessing the harm of patronizing adult entertainment clubs. International Journal of Public Opinion Research 18.4: 475-487.

Yang Liu, Elizabeth Shriberg, Andreas Stolcke, and Mary Harper. 2004. Using machine learning to cope with imbalanced classes in natural speech: Evidence from sentence boundary and disfluency detection. In Eighth International Conference on Spoken Language Processing.

Shirley D. McCune and Martha Matthews. 1978. Implementing Title IX and attaining sex equity: a workshop package for elementary-secondary educators: the community's role: outline and participants' materials for application sessions for community group members. Department of Health, Education, and Welfare,[Education Division], Office of Education.

Julia Mendelsohn, Yulia Tsvetkov, and Dan Jurafsky. 2020. A framework for the computational linguistic analysis of dehumanization.

Carla Pérez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 2020. Don't patronize me! An annotated dataset with patronizing and condescending language towards vulnerable communities. In Proceedings of the 28th International Conference on Computational Linguistics. pages 5891-5902.

Carla Pérez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 2022. SemEval-2022 Task 4: Patronizing and Condescending Language Detection. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022). Association for Computational Linguistics.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2019. Social bias frames: Reasoning about social and power implications of language. arXiv preprint arXiv:1911.03891.

Zijian Wang and Christopher Potts. 2019. Talkdown: A corpus for condescension detection in context. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the

9th International Joint Conference on Natural Language Processing

Bee Wah Yap, Khatijahhusna Abd Rani, Hezlin Aryani Abd Rahman, Simon Fong, Zuraida Khairudin, Nik Nairan Abdullah. 2014. An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. Proceedings of the first international conference on advanced data and information engineering (DaEng-2013). Springer, Singapore.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 Task 6: Identifying and categorizing offensive language in social media (offenseval). arXiv preprint arXiv:1903.08983.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Cagrı Coltekin. 2020. In Proceedings of SemEval Semeval-2020 task 12: Multilingual offensive language identification in social media.

Tuanfei Zhu, Yaping Lin, and Yonghe Liu. 2017. Synthetic minority oversampling technique for multiclass imbalance problems. Pattern Recognition, 72, 327-340.