UTSA NLP at SemEval-2022 Task 4: An Exploration of Simple Ensembles of Transformers, Convolutional, and Recurrent Neural Networks

Xingmeng Zhao and **Anthony Rios**
Department of Information Systems and Cyber Security
University of Texas at San Antonio
San Antonio, TX
{xingmeng.zhao, anthony.rios}@utsa.edu

Abstract

The act of appearing kind or helpful via the use of but having a feeling of superiority condescending and patronizing language can have have serious mental health implications to those that experience it. Thus, detecting this condescending and patronizing language online can be useful for online moderation systems. Thus, in this manuscript, we describe the system developed by Team UTSA SemEval-2022 Task 4, Detecting Patronizing and Condescending Language. Our approach explores the use of several deep learning architectures including RoBERTa, convolutions neural networks, and Bidirectional Long Short-Term Memory Networks. Furthermore, we explore simple and effective methods to create ensembles of neural network models. Overall, we experimented with several ensemble models and found that the a simple combination of five RoBERTa models achieved an F-score of .6441 on the development dataset and .5745 on the final test dataset. Finally, we also performed a comprehensive error analysis to better understand the limitations of the model and provide ideas for further research.

1 Introduction

Patronizing and condescending language (PCL) generally appears as an act to hold a superior attitude, resulting in language that "talks down" to others. For instance, PCL may describe someone in a power position as the potential "savior" of a vulnerable community (e.g., "*A donation of one dollar can save a life*"), masquerading a sense of superiority as compassion. There has been recent research suggesting that PCL can have adverse effects on the mental health of individuals (Giles et al., 1993; Shaw and Gordon, 2021), particularly in the context of ageism. While there has been substantial research on PCL in various contexts (Huckin, 2002; Komrad, 1983; Giles et al., 1993; Shaw and Gordon, 2021), unfortunately, there have been few efforts to develop PCL detectors in the field of Natural Language Processing (NLP). Hence, this paper describes our team's (UTSA NLP) contributions on the SemEval-2022 Task 4 (Pérez-Almendros et al., 2022) that introduced a new dataset for detecting PCL language.

NLP has investigated a broad spectrum of problematic language usages, such as hate speech (Vidgen et al., 2021), sarcasm language (Bamman and Smith, 2015), fake news (Hu et al., 2021), and the spread of rumors and disinformation. However, PCL has only recently been explored in the NLP community (**?**). To alleviate this issue, SemEval-2022 Task 4 expanded on the work by **?**, releasing a large PCL dataset for two PCL subtasks. Subtask 1 focuses on detecting the presence of PCL in news stories. PCL detection consists of a variety of sub-problems, for instance, identifying the exact PCL type expressed post (if any). There are multiple technical challenges for identifying PCL. For instance, accurate models must handle imbalanced data (most news stories do not contain PCL) and complex semantic understanding to relate shallow solutions for helping vulnerable populations. For instance, **?** describe "Shallow Solutions" as a type of PCL, e.g., "*Raise money to combat homelessness by curling up in sleeping bags for one night*". Nevertheless, for a model to understand this is an example of PCL, it would need to understand that "curling up in sleeping bags for one night" is unlikely to help the general problem of homelessness. Hence, we hypothesize that different models will learn to detect different types of PCL with varying accuracy; thus, combining multiple methods can result in better performance than a single method.

Overall, this paper describes our system for Task 1. Specifically, we evaluate multiple combined methods to handle PCL's complex nature better than a single method. Hence, for our methodology, we trained a RoBERTa model and two traditional deep learning models (a Convolutional

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 379 - 386

Neural Network and a Long Short-Term Memory Network) for comparison. In addition, we experimented with different model hyperparameters, random seeds, thresholds, and pre-trained word embeddings using the performance on the validation set to assess model variants. Finally, we evaluate multiple simple, yet effective, methods of combining the neural network models in an ensemble.

2 Background

Based on the work of **?**, the SemEval Task 4 dataset contains 10,637 news stories about vulnerable people published in 20 English-speaking countries, with a novel PCL taxonomy consisting of three top-level categories (The savior, The expert, and The poet) and seven low-level PCL categories describing the different types of condescension (Perez Almendros et al., 2020). Thet task contains two subtasks: binary classification (subtask 1) and multi-label classification (subtask 2). The binary classification for subtask 1 annotated the data with one of two categories: PCL and Not PCL. Subtask 2, the multi-label classification task, categorizes the news stories into a subset of of seven different PCL categories: unbalanced power relations, authoritative voice, shallow solution, presumption, compassion, metaphor, and the pooer, the merrier. A complete description of each category can be found in **?**.

PCL has been studied in a wide array of contexts, from sociolinguistics to healthcare (Huckin, 2002; Komrad, 1983; Giles et al., 1993; Shaw and Gordon, 2021). However, much of the prior work has focused on interviews and general qualitative methods. Thus, automated PCL detection models can provide social scientists with tools to understand the impact of PCL at scale. For instance, PCL models would allow linguistics to understand the implicit language actions related to condescension and aid social scientists in researching the link between condescension and other characteristics like gender or socioeconomic status because these superior attitudes and discourse of pity can routinize discrimination and make it less visible(Ng, 2007). However, much of the research on harmful language in NLP has concentrated on the explicit, offensive, and apparent phenomena like false news identification, trustworthiness prediction and fact-checking, modeling offensive language, both generic and community-specific (Vidgen et al., 2021; Zampieri et al., 2019; Schmidt and Wiegand, 2019); or how rumors spread (Ma et al., 2017). Recently, some work on condescending language has begun to surface. For example, based on the challenge that condescension is often undetectable from isolated discourse because it depends on discourse and social context, Wang and Potts (2019) introduces the task of modeling the phenomenon of condescension in direct communication from an NLP perspective and developing a dataset with annotated social media messages. Likewise, **?** also trained various baseline models to examine how existing NLP approaches perform in this task. Although they observe that recognizing PCL is achievable, it is still difficult. Hence, the work by **?** formed the basis of SemEval Task 4.

3 System Description

Overall, we developed an ensemble model strategy for the PCL challenge. Specifically, we evaluated three individual methods: RoBERTa, Convolutional Neural Networks, and Long Short-Term Memory Networks. Furthermore, we experimented with various ensemble combinations. Approaching the task we conduct multiple experiments with a variety neural network architectures using Convolutional Neural Networks (CNN) (Kim, 2014), Bi-directional long short term memory (BiLSTM) (Huang et al., 2015), and the pre-trained transformer-based model, RoBERTa (Liu et al., 2020). Each model and ensemble method is described in the section below.

CNN. We use the CNN model introduced by Kim (2014). Intuitively, the CNN model learns to extract predictive n-grams from the text. For the CNN architecture, we used filter sizes that span two, three, and four words. For the activation functions, we used ReLU (Glorot et al., 2011). Furthermore, we only needed two filters for each filter size [1]. Between the max-pooled outputs from the convolutional layer and the the full-connected output layer, we use dropout with the probability set to 0.5 during training. The final fully-connected output layer uses a Softmax activation and outputs class probabilities for PCL or Not PCL. The model was trained with the Adam optimizer (Kingma and Ba, 2015). Furthermore, we trained the CNN models with various learning rates randomly selected from 1e-4 to 1e-3 for a maximum of 35 epochs.

[1] We experimented with filter normal filter sizes from 100 to 300, but two seemed to perform just as well. We hypothesize this is because of the small number of PCL examples in the dataset.

BiLSTM. While CNNs only extract informative n-grams from text, recurrent neural networks (RNNs) are able to capture long term dependencies between words. For our RNN method, we use a Bidirectional Long Short-Term Memory Network (LSTM) (Hochreiter and Schmidhuber, 1997), specifically we use a variant introduced by Graves (2012). For the hyper-parameters, we did not use dropout, trained for a maximum of 35 epochs, and used variety hidden layer sizes (128, 256, and 512). The models were trained with the Adam optimizer (Kingma and Ba, 2015). Furthermore, we trained the BiLSTM models with various learning rates randomly selected from 1e-4 to 1e-3.

RoBERTa. In our study we used a variant of BERT (Kenton and Toutanova, 2019), namely RoBERTa (Liu et al., 2020) model, which is lighter and faster. Specifically, we use the roberta-base variant in the HuggingFace package (Wolf et al., 2019). We trained the RoBERTa model for 20 epochs with a mini-batch size set to 8 with the Adam optimizer. The learning rate was initially set to 2e-5 (other hyper-parameters same as (Liu et al., 2020)) and the adjusted stepwise linear decay was used to modify the learning rate through training, with step sizes of two and three used. Moreover, we used the last layer's CLS token which is passed to a final softmax layer. The model was check-pointed after each epoch, and the best version was chosen using the validation data.

Pre-trained Word Embeddings. For the CNN and BiLSTM models, we compare the following pre-trained word embeddings: Word2Vec vectors trained on Google News corpus (Mikolov et al., 2013), GloVe vectors trained on Wikipedia2014 and Gigaword5 corpus (GLoVe-Word) and Twitter (GLoVe-Twitter) corpora (Pennington et al., 2014), and FastText vectors trained on CommonCrawl corpora (Bojanowski et al., 2017).

Ensemble Model. There has been a wide array of research showing that ensembles of deep learning models have are useful for boosting model performance (Allen-Zhu and Li, 2020; Peng et al., 2018). We built different ensemble models by taking an unweighted average of the probability outputs of each of the independently trained models. This includes models trained with different hyperparameters, e.g., hidden state size for the BiLSTM models, different learning rates, and different random seeds. For the CNN and BiLSTM, each model was trained on four different pre-trained word vectors described above. The CNN and LSTM models were also trained on four different random seeds, for each combination of word embedding and learning rate. The RoBERTa model was trained with eleven random seeds and a number of two different step-wise learning rate schedulers using step sizes of two and three. Overall, we trained a total of 46 different models for the PCL task. Next, we experimented with two methods of model averaging (i.e., an ensemble): Ensemble 1 and Ensemble 2.

Model	Embedding	Seed	LR	HL
CNN	FastText	99	0.002	NA
LSTM	Glove_Twitter	99	0.002	128

Table 1: The hyperparameters for the best CNN and LSTM models found using random search. We report random seed (Seed), learning rate (LR), and pretrained embeddings (Embedding), and hidden layer size (HL) in this table.

First, for **Ensemble 1**, we simply average the top five model instances—which resulted in different RoBERTa models trained with different random seeds and learning step settings, i.e. step sizes of two with random seed of three; step size of three with random seed of zero, two, four, or seven. Second, for **Ensemble 2**, we experimented with taking the top five models combined with the top two CNN and BiLSTM models, and the hyperparameters (e.g., learning rate and embedding size) of the best performing models are shown in Table 1.

4 Dataset, Experimental Setup, and Training Details

For subtask 1, we use the train dataset provided by the PCL organizers. We choose the best epoch and the best hyperparameters using performance assessed in terms of F1-score on this development set based on the random search (Bergstra and Bengio, 2012). We saved the best epoch and best hyperparameters for each model variant. For evaluation, we use the provided test and validation datasets released by the organizers. We implemented our models on four GPUs using PyTorch (Paszke et al., 2019) to train binary classifiers for PCL. We use Cross-Entropy Loss in all our experiments as the loss function. We ran the experiments on a server using a GPU CUDA Version: 11.4. We selected the epoch based on the F1 score on the development set to save the best version of each model.

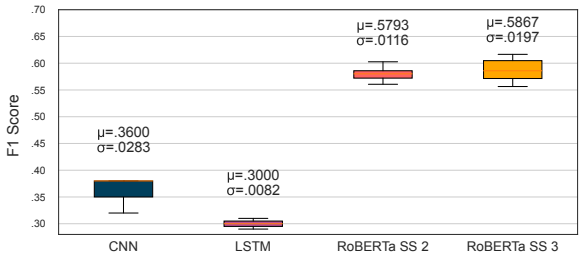

Figure 1: F1 score distributions for the best CNN, LSTM and RoBERTa models with the same hyperparameters, but different random seed values on the development dataset. SS stands for the learning rate step size.

To build a basis for comparison, all models were trained using the training data provided by the task organizers and evaluated against the provided validation dataset. The best-performing models were then submitted for evaluation against the test dataset during the task evaluation period. The training method was repeated three times for CNN and LSTM and eleven times for Roberta, each with a new random seed. This is because changing the random seed used in fine-tuning RoBERTa models can provide significantly different outcomes, even if the models are similar in terms of hyper-parameters (Dodge et al., 2020). The best-performing hyperparameters in each model were saved for the remainder of the ensemble study. We examined three random seeds (17, 42, and 99) for the best CNN and LSTM models, with standard deviations .0283 and .0082, respectively. We also evaluated ten random seeds for the RoBERTa model with step sizes of 2 and 3, yielding standard deviations for variance of .0116 and .0197, respectively. The distribution of each model's performance for different random seeds is shown in Figure 1.

We attempted to build a robust ensemble classifier with softmax output aggregation. For the ensemble model, the default threshold for interpreting probabilities as class labels is 0.5, but due to the imbalanced classification problem, we adjust the optimal threshold range from 0.1 to 0.9 when converting probabilities to class labels. We found the optimal probability threshold of CNN, LSTM, and RoBERTa that resulted in the best F1 score on the validation dataset were .1, .35, and .35, respectively.An optimal threshold was also chosen for the ensemble model, which was found to be .35.

		AVG P.	AVG R.	AVG F1
RoBERTa	step_size = 2	.5948	.5738	.5826
	step_size = 3	.6006	.5916	.5952
CNN	GoogleNews	.2542	.7085	.3733
	FastText	.2738	.5729	.3653
	Glove_Word	.2253	.4640	.3031
	Glove_Twitter	.2070	.6549	.3132
BiLSTM	GoogleNews	.3250	.4087	.3609
	FastText	.3659	.4020	.3810
	Glove_Word	.3525	.4271	.3831
	Glove_Twitter	.3745	.3953	.3821

Table 2: Average precision (AVG P.), average recall (AVG R.), and average F1 (AVG F1) for each model.

	step_size	seed	Prec.	Rec.	F1
Development Results					
RoBERTa	3	0	.6103	.6533	.6311
		4	.6263	.5980	.6118
		2	.6029	.6181	.6104
		7	.6277	.5930	.6098
	2	0	.5980	.6131	.6055
Ensemble 1	—	—	**.6215**	**.6683**	**.6441**
Ensemble 2	—	—	.6093	.6583	.6328
Test Results					
Ensemble 1	—	—	.5412	.5804	.5601
Ensemble 2	—	—	**.5599**	**.5899**	**.5745**

Table 3: Individual models in the best ensemble, and overall ensemble performance on the development and test datasets.

5 Results

Table 2 shows the average recall, precision, and F1 score. The scores are averaged across the different random seeds and hyperparameters used to train the models. Overall, we notice that the RoBERTa model outperforms both the CNN and BiLSTM models by more than 20%. For the CNN model, we find that the GoogleNews word embeddings perform best. However, for the BiLSTM model, we find that the model performs similarly across all pretrained embeddings, with the Glove Word embeddings slightly outperforming others.

In Table 3 we report the results of the two ensemble models: Ensemble 1 (only RoBERTa Models) and Ensemble 2 (Combining RoBERTa with the CNN and RNN models). On the development set, we find that that a single RoBERTa model achieves an F1 of .6311, with the next best four models achieving an F1 of around .61. The F1 of Ensemble 1 improves on the best RoBERTa models result

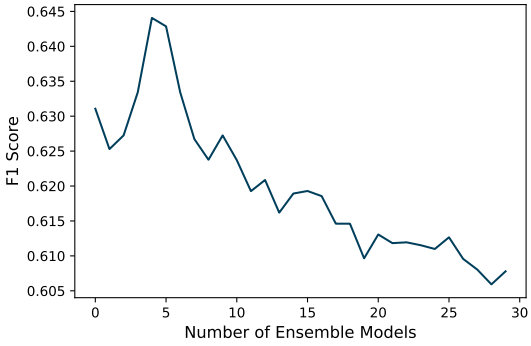

Figure 2: F1 score for different sized ensembles on the development dataset.

	FPs	FNs	Total PCL	Total
homeless	15	9	29	212
poor-families	13	17	38	190
women	3	8	14	233
in-need	15	2	33	226
immigrant	0	4	7	218
hopeless	15	9	26	217
vulnerable	6	4	20	209
migrant	2	3	5	207
disabled	4	7	14	194
refugee	8	3	13	188

Table 4: Summary of the false positives and false negatives found in each of the then PCL types.

by more than 1%. Ensemble 2 only improves on the Best F1 by .1%. However on the final test set, we find that the differences is not meaningful, with Ensemble 2 slightly outperforming Ensemble 1 (.56 vs. .57).

Next, in Figure 2 we report the results of averaging different number of models in our ensemble. More specifically, we evaluate averaging the best two models, best three models, and best N models, for N up to an ensemble of 30 different models. The model is chosen based on the top N performing models across all model types and random seeds. Overall, we see that initially the result of an ensemble of size 1 (i.e., only using the best RoBERTa model) has an F1 of around .63. However, that slowly increases beyond .64 at around the top five models. After that, the results slow decrease. Overall, we find that while a few models with varying performance improves the results. The more inaccurate models slowly outweigh the best performing model, thus decreasing the overall results. However, we find that the results stabilize around 0.61. Finally, in Table 4 we measure the number of False Positives and False Negatives for each of the main keywords identified in the PCL dataset (e.g., they keywords are provided by the organizers indicating a vulnerable group). The model produced 66 false negative predictions and 81 false positives predictions in total, but most of false positive errors are come from homeless,in-need, poor families, and hopeless. And the false negative error occur more frequently among the homeless, woman, immigrant, migrant and disabled topic.

5.1 False Positives and False Negatives

In addition to an exploration of the observation results, we perform an error analysis by manually comparing the true labels and predictions of Ensemble 1. First, for False Positives, we analyze an example related to "hopelessness/homelessness":

> **FP Example**:"The City Without Drugs organisation is still active , as is their YouTube channel . It features hundreds of videos of drug addicts being dragged half-conscious through the street , their faces not blurred , or confessing their alleged worthlessness , their hopelessness , their shame."

This paragraphs is predicted as PCL, but the ground-truth is Not PCL. This example indicates that the Ensemble incorrectly identifies sentences as PCL when they contain many PCL-related words that may be related to PCL-like text (e.g., related to hopelessness), even when the text is not directly indicating a feeling of superiority. Another false positive example is from the "homeless" topics:

> **FP Example**:"Viral photo helping fund homeless kid , his dog."

We can see that the entity of this sentence is a single individual. This paragraph is recognized as PCL by the system, maybe because the PCL system believes it contains the shallow solution (i.e., viral photo). However, it neglected the fact that financing a specific homeless child may be realistic, i.e. it may not be a shallow solution for a single person. Perez Almendros et al. (2020) also mention that shallow solutions are also often overlooked by RoBERTa, where recognizing shallow solutions in the text requires external knowledge of the situation and the needs of those affected. Thus, a large number of false positive results are generated

by misidentifying the entities and the relationship between patronizing and condescending language. Next, we look at a False Negative:

FN Example: "Charity plans to forgo parking so homeless can have gym and medical centre."

Here we find another issue with shallow solutions. Specifically, the model is not able to associate the proposed procedure as not being a method of really addressing homelessness. Specifically, the PCL system unaware that "forgoing parking" is not a complete solution to help homeless people, which is a simple and superficial philanthropic effort that is unlikely to make a significant difference on vulnerable communities. The second example of a false negative concerns presuppositions. People need to decide whether the assumption made is reasonable or not for this type of PCL. We found for the political topic, like immigrant and migrant topics, there are many preconceived assumptions in this kind of news. For example, in the following situation, the author assumes Filipino families are poor and need assistance based on stereotypes.

FN Example:"But if the Supreme Court gives a favorable decision for the president , his immigration program would immediately take effect , changing the lives of eligible Filipino families and other immigrants."

This error suggests that the model is incapable of understanding complex relationships between vulnerable communities and ideas. A future interesting research avenue would explore methods for incorporating relevant knowledge bases, similar to recent work on common sense generation (Xing et al., 2021), into transformer models to address these errors.

6 Conclusion

In this paper, we have presented our submission for the PCL detection system submitted to the SemEval-2022 Task 4. Our team are focus on the subtask1 to identify whether the paragraphs contain the PCL or not. We proposed several ensemble models that leverages pre-trained word vectors and three different deep learning architectures. In future efforts, we plan to further improve our model by incorporating structured knowledge bases.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1947697.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *CoRR*, abs/2012.09816.

David Bamman and Noah Smith. 2015. Contextualized sarcasm detection on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, pages 574–577.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Howard Giles, Susan Fox, and Elisa Smith. 1993. Patronizing the elderly: Intergenerational evaluations. *Research on Language and Social Interaction*, 26(2):129–149.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Alex Graves. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 754–763.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

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

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Mark S Komrad. 1983. A defence of medical paternalism: maximising patients' autonomy. *Journal of medical ethics*, 9(1):38–44.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Sik Hung Ng. 2007. Language-based discrimination: Blatant and subtle forms. *Journal of Language and Social Psychology*, 26(2):106–122.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Yifan Peng, Anthony Rios, Ramakanth Kavuluru, and Zhiyong Lu. 2018. Extracting chemical–protein relations with ensembles of svm and deep learning models. *Database*, 2018.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Carla Perez 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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Anna Schmidt and Michael Wiegand. 2019. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain*, pages 1–10. Association for Computational Linguistics.

Clarissa A Shaw and Jean K Gordon. 2021. Understanding elderspeak: An evolutionary concept analysis. *Innovation in aging*, 5(3):igab023.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682.

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 (EMNLP-IJCNLP)*, pages 3711–3719, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yiran Xing, Zai Shi, Zhao Meng, Gerhard Lakemeyer, Yunpu Ma, and Roger Wattenhofer. 2021. Km-bart: Knowledge enhanced multimodal bart for visual commonsense generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 525–535.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. 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 1415–1420.