ULFRI at SemEval-2022 Task 4: Leveraging uncertainty and additional knowledge for patronizing and condescending language detection

Matej Klemen **Marko Robnik-Šikonja**
University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, Ljubljana, Slovenia
{matej.klemen, marko.robnik}@fri.uni-lj.si

Abstract

We describe the ULFRI system used in the Subtask 1 of SemEval-2022 Task 4 Patronizing and condescending language detection. Our models are based on the RoBERTa model, modified in two ways: (1) by injecting additional knowledge (coreferences, named entities, dependency relations, and sentiment) and (2) by leveraging the task uncertainty by using soft labels, Monte Carlo dropout, and threshold optimization. We find that the injection of additional knowledge is not helpful but the uncertainty management mechanisms lead to small but consistent improvements. Our final system based on these findings achieves $F_1 = 0.575$ in the online evaluation, ranking 19th out of 78 systems.

1 Introduction

Despite invaluable contributions to the society, the internet can also serve as an infrastructure for a rapid spread of hurtful language, in part due to the anonymity it commonly provides (Burnap and Williams, 2015). The spread of such language can have a serious impact on individuals, such as the increased development of mental health problems in children (Munro, 2011). To prevent this, the society has to establish moderation mechanisms. Fully manual content moderation is infeasible both due to the large scale of the web as well as the possible negative psychological effects on human moderators (Arsht and Etcovitch, 2018). Much attention has been devoted to the automatic detection of offensive language within the field of natural language processing (NLP). Some examples include the detection of hate speech (Davidson et al., 2017), toxic language (Pavlopoulos et al., 2021), and cyberbullying (Dadvar et al., 2013), which use a relatively explicit form of hurtful language (Waseem et al., 2017). In contrast, patronizing and condescending language (PCL) is more implicit in nature. PCL can roughly be described as an expression of a superior attitude towards others, possibly unconsciously. Perez Almendros et al. (2020) have shown that large language models are able to detect PCL to a various degree, but consistently better than random guessing or a machine learning approach using the bag-of-words representation. Based on that, the authors propose SemEval-2022 Task 4 (Pérez-Almendros et al., 2022), which aims to encourage further research and improvements in the detection of PCL.

We present our attempts at modeling PCL, based on the RoBERTa model (Liu et al., 2019) and following two main lines:

1. **Injection of additional knowledge**. We experiment with the injection of additional knowledge on coreferences, named entities, dependency relations, and sentiment.

2. **Leveraging uncertainty present in the task**. We experiment with the use of soft labels in the form of the target label probability distribution, and with the Monte Carlo dropout as a means for more accurate estimation of the label posterior probability (Gal and Ghahramani, 2016).

Our first set of modifications aims to guide the model to better follow the definition of PCL. The additional coreference and named entity knowledge may help the model to focus on detecting an imbalance between entities in the text, while the dependency relations and sentiment knowledge may help the model discover more subtle linguistic patterns used in PCL. We inject different forms of knowledge as the second input sequence to the model, combining it with the primary text representation during training of the PCL detector. This is motivated by the Factored Transformer (Armengol-Estapé et al., 2021).

The second set of modifications aims to capture the subjectivity and uncertainty that is inherently present in the task and is reflected in the annotator disagreement. This is motivated by the data per-

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 525 - 532
July 14-15, 2022 ©2022 Association for Computational Linguistics

spectivism paradigm (Basile et al., 2021) which argues the disagreements are not necessarily errors.

We participate in Subtask 1, and our best model ranks 19th out of 78 systems[1]. In our analysis, we find that (1) injection of additional knowledge does not increase the F_1 score significantly and (2) leveraging uncertainty in the task leads to small but consistent increase in the F_1 score.

The remainder of the paper is structured as follows. In Section 2, we describe the details of the task. In Section 3, we describe our approach, and analyze its performance in Section 4. In Section 5, we summarize our work and provide ideas for further work.

2 Task Description

Given an updated version of the *Don't Patronize Me!* dataset (Perez Almendros et al., 2020), the goal of SemEval-2022 Task 4 Subtask 1 is the detection of patronizing and condescending language (PCL). The provided dataset consists of 10 469 paragraphs annotated by three annotators: two were tasked with annotating the examples as not containing PCL (0), containing PCL (2), or borderline (1). The third annotator resolved *complete* disagreements, i.e. examples annotated as $\{0, 2\}$ by the two annotators. The annotations are aggregated into a five-point fine-grained class y_F:

- $y_F = 0$ if both annotators assigned the label 0,
- $y_F = 1$ if one annotator assigned the label 0 and the other assigned 1,
- $y_F = 2$ if both annotators assigned the label 1,
- $y_F = 3$ if one annotator assigned the label 2 and the other assigned 1,
- $y_F = 4$ if both annotators assigned the label 1.

We provide the distribution of these fine-grained class labels in Figure 1. For the task evaluation, the fine-grained five label class is binarized into a coarse-grained class y_C, where $y_C = 1$ if $y_F \in \{2, 3, 4\}$, and $y_C = 0$ otherwise. Although the final evaluation uses binary labels, the fine-grained labels can provide additional information in the form of label uncertainty. We leverage this information in one of our modifications, described next.

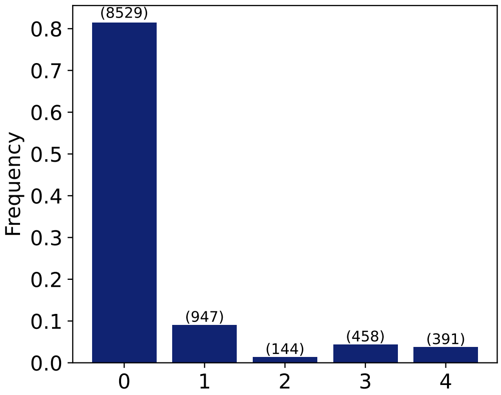

Figure 1: Fine-grained PCL label distribution. The numbers above bars indicate the number of examples for each label.

3 Methods

In this section, we describe our methodology. First, we describe RoBERTa, which we use as the baseline model. Then, we describe how additional knowledge is injected into the model in Section 3.2. In Section 3.3 we describe how we leverage the task uncertainty in our model.

3.1 RoBERTa

RoBERTa (Liu et al., 2019) is a robustly optimized BERT model (Devlin et al., 2019), composed of multiple transformer layers that use the self-attention mechanism to construct a text representation. It is first pre-trained on a general corpus using the masked language modeling objective, after which it can be fine-tuned for a downstream task. Motivated by its strong performance on the PCL detection task shown by Perez Almendros et al. (2020), we use the RoBERTa$_{\text{BASE}}$ model as a baseline.

3.2 Knowledge injection

We inject various types of additional knowledge through a secondary aligned input sequence containing additional knowledge in the form of special tokens. The procedure is shown in Figure 2 for one type of additional knowledge. Using RoBERTa, we independently obtain two representations and combine them using a learned weighted linear combination to obtain a single representation. Lastly, a linear layer transforms the representation into label

[1] Our code is available at https://github.com/matejklemen/pcl-detection-semeval2022t4.

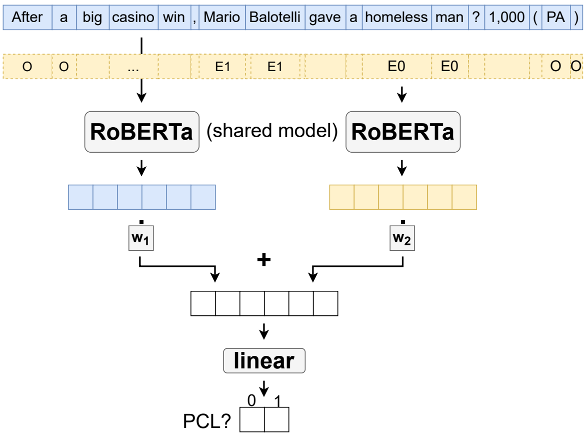

Figure 2: Injection of additional coreference knowledge for PCL detection. The secondary sequence (in yellow) consists of special tokens that denote if a word represents an entity (E_i) or not (O).

scores. The individual sequence representations correspond to the output of the last layer for the <s> token in each sequence.

We experiment with four different types of additional knowledge, one at a time: coreferences (obtained using neuralcoref[2]), sentiment (obtained using SentiWordNet (Esuli and Sebastiani, 2006)), named entities, and dependency relations (obtained using Stanza (Qi et al., 2020)). We provide additional preprocessing details and the used tagsets in Appendix A.

As our modification requires embedding two input sequences instead of one, the memory requirement during training is doubled, and the batch size has to be halved. To minimize differences due to a halved batch size, we accumulate gradients over two half-sized batches before updating the parameters.

3.3 Leveraging uncertainty

We experiment with two ways to leverage the task uncertainty. The first approach trains a model on soft instead of hard (one-hot encoded) labels. We show the comparison between hard and soft labels in Table 1. As described in Section 2 each example is annotated twice. We assign each annotation a probability of the example containing PCL: 0.0 if the annotation is 0, 1.0 if it is 2, and 0.5 if it is 1 (borderline). To obtain the final soft labels, we then take the mean of the two annotations. In this way, we transform a five-class problem into a binary one while approximately preserving information about label differences. Additionally, we potentially avoid issues when a label has few training examples.

Table 1: Conversion scheme from fine-grained annotations into hard and soft binary target vector.

Label type	Fine-grained annotation (y_F)	Binary target vector
hard	0, 1	[1.00, 0.00]
	2, 3, 4	[0.00, 1.00]
soft	0	[1.00, 0.00]
	1	[0.75, 0.25]
	2	[0.50, 0.50]
	3	[0.25, 0.75]
	4	[0.00, 1.00]

The second approach uses the Monte Carlo dropout (MCD) (Gal and Ghahramani, 2016) to sample the label distribution during the prediction phase. Instead of determining the target label using a single prediction, we obtain multiple non-deterministic predictions while applying dropout (Srivastava et al., 2014), and then aggregate them into a single prediction (in our case, using the mean) (Miok et al., 2022).

Both modifications transform the target label probability distribution, so using the PCL probability threshold of 0.5 may no longer be suitable. For this reason, we also experiment with the decision threshold optimization, i.e. we select the threshold based on the validation set F_1 score.

4 Evaluation

In this section, we evaluate our methodology and compare it to the baseline. We start by describing the experimental settings in Section 4.1, and continue with the results in Section 4.2.

4.1 Experimental settings

We select the hyperparameters for the training of RoBERTa using the validation set F_1 score in preliminary experiments on a single 80%:10%:10% split into the training, validation and testing set. In our main experiments, we use the learning rate 10^{-5}, maximum sequence length of 158, and batch size of 48. The latter two were selected in a way to allow training on an 11GB GPU.

In the evaluation, we use 10-fold cross validation and report the means across folds. In each

[2] https://github.com/huggingface/neuralcoref

cross validation iteration, we use 10% of the training set for tuning and early stopping. Following the official evaluation, we use three metrics: precision, recall, and F_1 score for the positive (PCL) label. To improve clarity, we only report the mean F_1 score throughout this section, and provide other metrics and standard deviations of the results in Appendix B. We statistically test the differences in F_1 score between pairs of models using the Wilcoxon signed-rank test (Wilcoxon, 1945). The same test is applied to the difference between groups of models, where one group uses and one group does not use certain modification (e.g., soft labels). In all cases, we use a confidence level $\alpha = 0.01$ to determine the significance of the differences.

For the online evaluation, we retrain the model using the best parameters on a 90%:10% split into a training and validation set.

4.2 Results

Table 2 shows the F_1 scores of our enhanced models in comparison to the RoBERTa$_{\text{BASE}}$ baseline. We interpret the results below, starting with knowledge-enhanced models in Section 4.2.1 and models leveraging uncertainty in Section 4.2.2.

4.2.1 Knowledge-enhanced models

We first only consider the knowledge injection in isolation, i.e. the scores for each type of knowledge in Table 2a.

We can observe that the addition of knowledge about coreferences ($F_1 = 0.575$), named entities ($F_1 = 0.563$), and dependency relations ($F_1 = 0.567$) increases the performance over the baseline ($F_1 = 0.556$). However, none of the increases are statistically significant due to the variance in performance across the folds.

4.2.2 Knowledge- and uncertainty-enhanced models

Next, we consider the effect of leveraging uncertainty both on the base model as well as the knowledge-enhanced models, i.e. analyze the full results in Table 2. Unless stated otherwise, we compare the modified models with their respective base model without the discussed modifications (i.e. not necessarily always against the roberta-base model without MCD).

The first observation is that training the models on soft instead of hard labels in most cases improves the F_1 score both when not using MCD and when using MCD. Using soft labels increases the

Table 2: Results of the base models and their modifications. The best score in each table is displayed in bold.

(a) Results without MCD.

	train on hard labels	train on soft labels
Model	$\mathbf{F_1}$	$\mathbf{F_1}$
roberta-base	0.556	0.570
+ opt. thresh.	0.550	0.577
+ coreference	0.575	**0.582**
+ opt. thresh.	0.573	0.572
+ sentiment	0.544	0.554
+ opt. thresh.	0.532	**0.582**
+ named ent.	0.563	0.568
+ opt. thresh.	0.563	0.577
+ dep. relations	0.567	0.580
+ opt. thresh.	0.557	0.578

(b) Results using 10 MCD rounds.

	train on hard labels	train on soft labels
Model	$\mathbf{F_1}$	$\mathbf{F_1}$
roberta-base	0.546	0.553
+ opt. thresh.	0.551	0.580
+ coreference	0.550	0.566
+ opt. thresh.	0.573	0.579
+ sentiment	0.526	0.535
+ opt. thresh.	0.540	0.575
+ named ent.	0.556	0.556
+ opt. thresh.	0.553	0.577
+ dep. relations	0.557	0.557
+ opt. thresh.	0.564	**0.583**

(c) Results using 50 MCD rounds.

	train on hard labels	train on soft labels
Model	$\mathbf{F_1}$	$\mathbf{F_1}$
roberta-base	0.546	0.554
+ opt. thresh.	0.556	**0.586**
+ coreference	0.549	0.570
+ opt. thresh.	0.568	0.580
+ sentiment	0.527	0.563
+ opt. thresh.	0.543	0.581
+ named ent.	0.557	0.563
+ opt. thresh.	0.569	0.581
+ dep. relations	0.554	0.560
+ opt. thresh.	0.569	0.577

F_1 score for 13 models and makes no difference for 2 models (named entity and dependency relation enhanced models using 10 MCD rounds). The differences are statistically significant.

Optimizing the threshold after training on soft labels improves the F_1 score for 14 models: for 13 models, the improvement over baseline is larger than without threshold optimization. The only exception where the F_1 score decreases is the coreference enhanced model without MCD, which could be due to overfitting the threshold to the tuning set. Optimizing the threshold after training on hard labels has a mixed effect. For models without MCD, it consistently leads to equivalent or lower F_1 scores compared to the baseline model without the optimized threshold. On the other hand, the threshold optimization leads to the statistically significant increase in the F_1 score for models with MCD. Concretely, it increases the F_1 score for 9 models and decreases it for 1 model.

Using MCD on its own is not helpful. Without also training the model on soft labels or optimizing the threshold, it decreases the F_1 score for all 10 models in comparison to the respective models not using MCD. However, as described previously, using MCD in combination with either or both of these mechanisms mostly leads to an increase in F_1 score.

Lastly, using more MCD rounds starts to bring a diminishing increase in the F_1 score after a certain point. Concretely, using 50 instead of 10 MCD rounds leads to only a small additional increase in F_1 score.

The best F_1 score is achieved by the model without additional knowledge, trained on soft labels, using 50 MCD rounds and the threshold optimization. This model achieves the F_1 score of 0.586 ($+0.030$ over the baseline) in the offline evaluation, and the F_1 score 0.575 on the official online test set.

5 Conclusion

We have described our approaches for the detection of PCL as part of the SemEval-2022 Task 4. We attempted to inject knowledge into prediction models and leverage the uncertainty present in the task. The injection of additional knowledge did not increase the F_1 score significantly. Leveraging the uncertainty in different ways produced mixed effects. Training the models on soft instead of hard labels consistently increased the F_1 score, while using MCD on its own was not beneficial. However, using MCD in combination with soft labels and threshold optimization brought consistent improvements in the F_1 score and produced our best score.

Both directions of our research have potential for further work. In our knowledge injection experiments, we have only experimented with a single type of additional knowledge at a time. To inject multiple types simultaneously, we would need to create special tokens for each combination, which could lead to overfitting due to relatively small and imbalanced data. Therefore, in further work we will try a different method for knowledge injection considering multiple types of additional knowledge simultaneously. In leveraging uncertainty, we have constructed the soft binary labels from the two annotations per example and aggregated the annotations by weighing them equally. A possible further work would experiment with different weighting schemes.

Acknowledgements

The research was supported by the Slovene Research Agency through research core funding no. P6-0411, project J6-2581 (CANDAS - Computer-assisted multilingual news discourse analysis with contextual embeddings), and the young researcher grant.

References

Jordi Armengol-Estapé, Marta R. Costa-jussà, and Carlos Escolano. 2021. Enriching the transformer with linguistic factors for low-resource machine translation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 73–78.

Andrew Arsht and Daniel Etcovitch. 2018. The human cost of online content moderation. *Harvard Journal of Law and Technology Digest*.

Valerio Basile, Federico Cabitza, Andrea Campagner, and Michael Fell. 2021. Toward a perspectivist turn in ground truthing for predictive computing. In *Conference of the Italian Chapter of the Association for Intelligent Systems (ItAIS 2021)*.

Pete Burnap and Matthew L. Williams. 2015. Cyber hate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2):223–242.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *Advances in Information Retrieval*, pages 693–696.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, pages 512–515.

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.

Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kristian Miok, Blaž Škrlj, Daniela Zaharie, and Marko Robnik-Šikonja. 2022. To BAN or not to BAN: Bayesian attention networks for reliable hate speech detection. *Cognitive Computation*, 14(1):353–371.

Emily Munro. 2011. The protection of children online: A brief scoping review to identify vulnerable groups. Accessed: January 27, 2022.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043.

John Pavlopoulos, Jeffrey Sorensen, Léo Laugier, and Ion Androutsopoulos. 2021. SemEval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 59–69.

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.

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)*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Table 3: Used tagsets for representing additional knowledge.

Knowledge	Tagset
coreference	O, ENT0, ENT1, . . . , ENT50; 52 tags
sentiment	NEG, OBJ, POS, UNK; 4 tags
named ent.	O, {B-, I-, E-, S-} × {ORG, PER, LOC, MISC}; 17 tags
dep. relations	universal dep. relations (including subtypes) (Nivre et al., 2020); 63 tags

A Additional details of knowledge injection experiment

Table 3 shows the tags used to inject the additional knowledge. The tags are added as special (indivisible) tokens to the tokenizer and used in the secondary input sequence as described in Section 3.2. For sentiment and coreference knowledge, we note additional preprocessing details:

- **Sentiment.** We obtain the sentiment tags using SentiWordNet. Each token is assigned a positive, negative and objectivity score, and

Table 4: Extended results of the base models and their modifications: mean and standard deviation of precision (P), recall (R), and F_1 score across 10 folds.

(a) Results without MCD.

	train on hard labels			train on soft labels		
Model	**P**	**R**	$\mathbf{F_1}$	**P**	**R**	$\mathbf{F_1}$
roberta-base	0.575 (0.051)	0.543 (0.057)	0.556 (0.038)	0.612 (0.040)	0.539 (0.067)	0.570 (0.045)
+ opt. thresh.	0.571 (0.056)	0.537 (0.068)	0.550 (0.048)	0.597 (0.060)	0.570 (0.081)	0.577 (0.051)
+ coreference	0.591 (0.046)	0.567 (0.063)	0.575 (0.032)	0.583 (0.073)	0.594 (0.061)	0.582 (0.031)
+ opt. thresh.	0.566 (0.061)	0.593 (0.066)	0.573 (0.033)	0.572 (0.084)	0.591 (0.077)	0.572 (0.028)
+ sentiment	0.589 (0.056)	0.514 (0.074)	0.544 (0.054)	0.614 (0.061)	0.529 (0.132)	0.554 (0.077)
+ opt. thresh.	0.587 (0.043)	0.494 (0.080)	0.532 (0.056)	0.596 (0.061)	0.585 (0.097)	0.582 (0.047)
+ named ent.	0.538 (0.053)	0.604 (0.076)	0.563 (0.026)	0.569 (0.067)	0.585 (0.081)	0.568 (0.028)
+ opt. thresh.	0.563 (0.021)	0.588 (0.087)	0.563 (0.021)	0.565 (0.044)	0.595 (0.062)	0.577 (0.031)
+ dep. relations	0.537 (0.028)	0.605 (0.076)	0.567 (0.040)	0.581 (0.042)	0.585 (0.064)	0.580 (0.028)
+ opt. thresh.	0.572 (0.061)	0.558 (0.091)	0.557 (0.039)	0.593 (0.037)	0.575 (0.085)	0.578 (0.038)

(b) Results using 10 MCD rounds.

	train on hard labels			train on soft labels		
Model	**P**	**R**	$\mathbf{F_1}$	**P**	**R**	$\mathbf{F_1}$
roberta-base	0.610 (0.062)	0.499 (0.054)	0.546 (0.041)	0.660 (0.054)	0.481 (0.068)	0.553 (0.056)
+ opt. thresh.	0.580 (0.064)	0.534 (0.069)	0.551 (0.040)	0.593 (0.043)	0.577 (0.095)	0.580 (0.059)
+ coreference	0.612 (0.043)	0.507 (0.084)	0.550 (0.051)	0.614 (0.069)	0.537 (0.075)	0.566 (0.038)
+ opt. thresh.	0.537 (0.058)	0.621 (0.055)	0.573 (0.040)	0.572 (0.078)	0.602 (0.085)	0.579 (0.043)
+ sentiment	0.627 (0.063)	0.460 (0.074)	0.526 (0.059)	0.667 (0.075)	0.474 (0.129)	0.535 (0.086)
+ opt. thresh.	0.575 (0.056)	0.514 (0.058)	0.540 (0.047)	0.610 (0.059)	0.553 (0.077)	0.575 (0.042)
+ named ent.	0.589 (0.062)	0.543 (0.087)	0.556 (0.034)	0.621 (0.077)	0.523 (0.098)	0.556 (0.041)
+ opt. thresh.	0.560 (0.062)	0.561 (0.086)	0.553 (0.034)	0.540 (0.056)	0.628 (0.074)	0.577 (0.043)
+ dep. relations	0.577 (0.039)	0.544 (0.063)	0.557 (0.037)	0.633 (0.062)	0.511 (0.100)	0.557 (0.060)
+ opt. thresh.	0.555 (0.044)	0.577 (0.046)	0.564 (0.033)	0.572 (0.059)	0.608 (0.076)	0.583 (0.031)

(c) Results using 50 MCD rounds.

	train on hard labels			train on soft labels		
Model	**P**	**R**	$\mathbf{F_1}$	**P**	**R**	$\mathbf{F_1}$
roberta-base	0.609 (0.055)	0.499 (0.051)	0.546 (0.038)	0.662 (0.052)	0.480 (0.062)	0.554 (0.051)
+ opt. thresh.	0.564 (0.046)	0.558 (0.090)	0.556 (0.046)	0.594 (0.058)	0.587 (0.063)	0.586 (0.034)
+ coreference	0.616 (0.035)	0.503 (0.078)	0.549 (0.046)	0.620 (0.080)	0.538 (0.072)	0.570 (0.043)
+ opt. thresh.	0.561 (0.084)	0.601 (0.099)	0.568 (0.040)	0.582 (0.084)	0.599 (0.101)	0.580 (0.048)
+ sentiment	0.630 (0.066)	0.461 (0.074)	0.527 (0.061)	0.629 (0.080)	0.531 (0.099)	0.563 (0.045)
+ opt. thresh.	0.574 (0.040)	0.523 (0.092)	0.543 (0.055)	0.543 (0.058)	0.633 (0.059)	0.581 (0.038)
+ named ent.	0.589 (0.068)	0.544 (0.086)	0.557 (0.038)	0.629 (0.080)	0.531 (0.099)	0.563 (0.045)
+ opt. thresh.	0.551 (0.038)	0.596 (0.072)	0.569 (0.030)	0.543 (0.058)	0.633 (0.059)	0.581 (0.038)
+ dep. relations	0.575 (0.033)	0.541 (0.071)	0.554 (0.043)	0.636 (0.056)	0.513 (0.091)	0.560 (0.049)
+ opt. thresh.	0.556 (0.043)	0.588 (0.062)	0.569 (0.035)	0.568 (0.058)	0.607 (0.107)	0.577 (0.050)

the final tag is the one with the highest of the three scores. The token UNK is used if a token does not have associated scores.

- **Coreference.** We enumerate the coreference clusters in the document randomly. We find this has a positive effect on the performance, possibly as the model is overtrained on the tags with lower IDs otherwise.

B Extended evaluation results

Table 4 shows the extended results of the base and extended models.