ReDASPersuasion at SemEval-2023 Task 3: Persuasion Detection using Multilingual Transformers and Language Agnostic Features

Fatima Zahra Qachfar
University of Houston
Houston, TX, USA
fqachfar@uh.edu

Rakesh M. Verma
University of Houston
Houston, TX, USA
rmverma2@central.uh.edu

Abstract

This paper describes a multilingual persuasion detection system that incorporates persuasion technique attributes for a multi-label classification task. The proposed method has two advantages. First, it combines persuasion features with a sequence classification transformer to classify persuasion techniques. Second, it is a language agnostic approach that supports a total of 100 languages, guaranteed by the multilingual transformer module and the Google translator interface. Our persuasion system outperforms the SemEval baseline in all languages, except zero shot prediction languages, which was not the main focus of our research. With the highest F1-Micro score of 0.45 for Italian, it achieved the eighth position on the leaderboard.

1 Introduction

The goal of the SemEval-2023 Task 3 (Piskorski et al., 2023) Subtask 3 is to detect the presence of one or many persuasion techniques in text data. It is a multilingual task and consisted initially of six languages: English, French, Polish, Italian, Russian and German. In the testing phase, organizers added three surprise languages (Spanish, Greek, and Georgian) for zero-shot classification tasks.

In this work, we aim to build a cross-lingual predictive model that detects persuasion techniques in a given text with the help of language transformers and linguistic features. Our system uses XLM-RoBERTa (Conneau et al., 2019) as a backbone model, incorporating language agnostic features provided using Google translation over 100 languages for persuasion detection. XLM-RoBERTa has been shown to be a powerful multilingual pretrained language model compared against other models like Multilingual BERT (M-BERT) (Devlin et al., 2018) and DistilBERT (Sanh et al., 2019). Also, this language transformer can process all the languages existing in the SemEval-2023 Task3.

Through this task, we realized the importance of data quality in Multi-label classification tasks with cross-lingual settings. We were also able to build a new system that incorporates language agnostic features to capture persuasion in text semantics, accompanied by the powerful capacity of language transformers based on attention mechanism.

2 Background

2.1 Task Description

The SemEval-2023 Task 3 (Piskorski et al., 2023) had three subtasks: (i) Subtask 1 is news genre categorization, (ii) Subtask 2 is framing detection, and (iii) Subtask 3 is persuasion techniques detection. We focus solely on the multi-label classification problem Subtask 3 that aims to detect 23 persuasion techniques. All subtasks are multilingual and contain zero-shot learning.

In Subtask 3, each article in the dataset is labeled with multiple classes. One article can contain different persuasion techniques (persuasion data statistics are listed in Table 6, Appendix B).

2.2 Task Evaluation

Subtask 3 is evaluated using the Micro F1 score between the gold labels and model predictions defined in Equation (1).

$$F1\_Micro = \frac{TP}{TP + \frac{FP+FN}{2}} \quad (1)$$

F1-Micro is based on True Positives (TP), False Positives (FP) and False Negatives (FN), instead of individually for each class, which makes it well suited for multi-label classification.

3 System Overview

In this section, we describe our model and the steps in our approach. Figure 1 depicts the architecture of the entire *ReDASPersuasion* system.

Our System consists of three modules: (i) The first module is a multilingual transformer model, (ii) The second feature engineering module focuses

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 2124–2132
July 13-14, 2023 ©2023 Association for Computational Linguistics

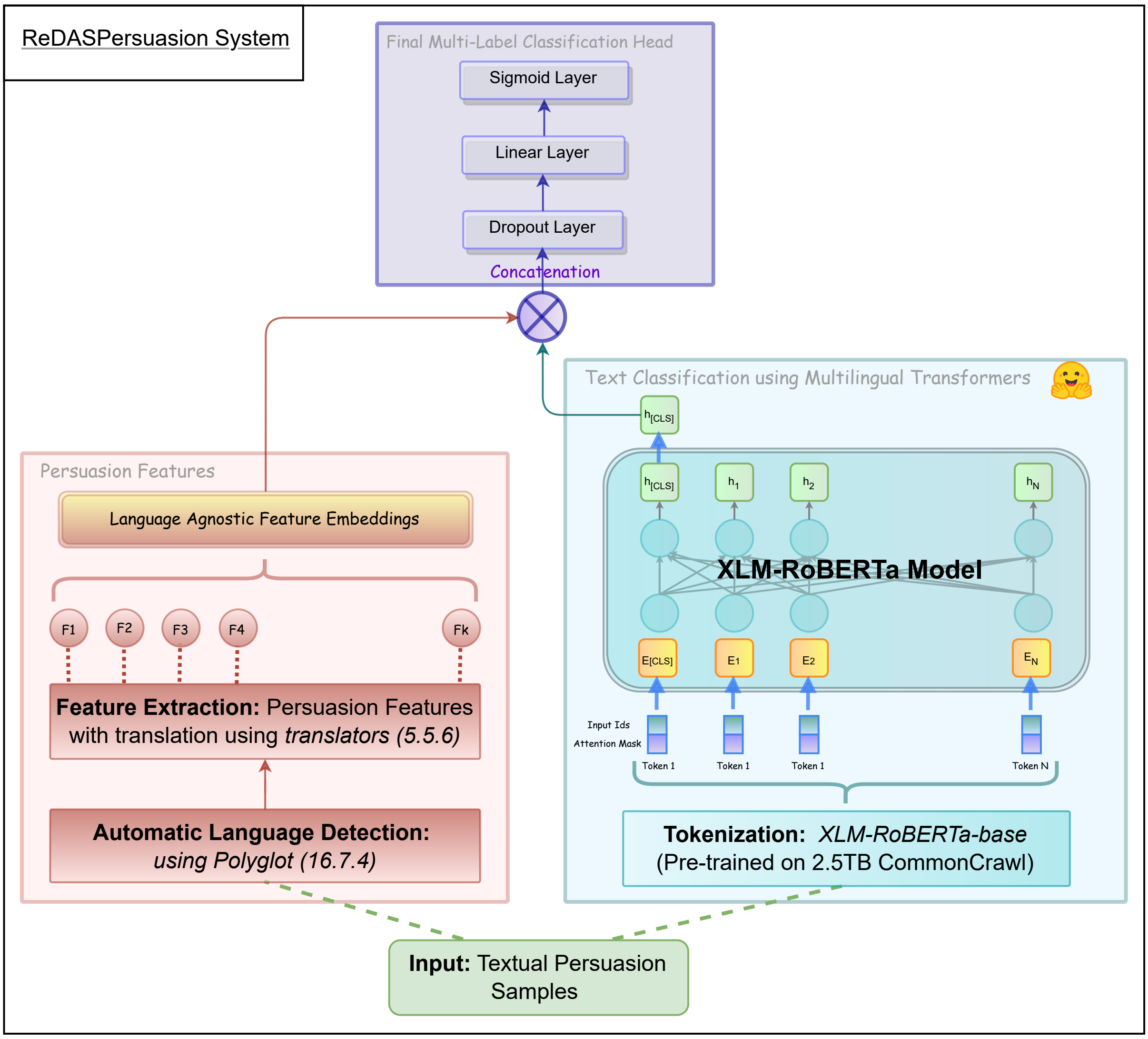

Figure 1: Our Persuasion Detection System Architecture

on building language agnostic features for cross-lingual classification of persuasion techniques, and (iii) The third final module is a multi-label classification head where the *[CLS]* embeddings from transformer classification are combined with persuasion features to feed this information to a dropout layer, followed by a dense linear layer, and finally a sigmoid activation function in order to perform multi-label classifications.

Initially, we concatenate the *[CLS]* embeddings with a total of six persuasion features that we will introduce in the next section. We also use a dropout of 0.3 to prevent overfitting. The dropout layer has the convenient property of speeding up training since fewer weights are required in each forward pass.

In our approach, the final layer consists of k independent sigmoid activation outputs ($k = 19$ for English and $k = 23$ for all other languages). The persuasion label prediction is based on a threshold value of the logit outputs, fine-tuned using the development set.

3.1 Persuasion Feature Engineering

We created task-specific and language agnostic features to detect most of the persuasion techniques within the scope of 100 languages. We will explain below the process we followed to engineer each feature individually.

3.1.1 Appeal to Fear

For this persuasion technique, we lean towards sentiment analysis to find if fear and anger are represented in input text. Polyglot (Al-Rfou et al., 2013) has polarity lexicons for over 136 languages to detect positive, negative and neutral text (Chen and

Skiena, 2014).

We extract **polarity** as the first persuasion feature because fear has a strong negative emotional inclination. In fact, polarity lies between [-1,1], where -1 defines a negative sentiment and 1 defines a positive sentiment.

Moreover, the "*Slogans*" persuasion technique also tends to act as emotional appeals. Therefore, using sentiment analysis is very important, and combines the scope of multiple persuasion techniques (Fear, Loaded Language, Slogans, Exaggeration).

3.1.2 Exaggeration and Minimization

We automatically analyze the text to find any Extreme Case Formulations (ECF) defined by (Mora, 2009) that invoke extreme descriptions of events or objects based on POS Tags, adverb wordlists and hyperboles. A simple example of an ECF is a sentence that contains an extreme description via an adjective or intensified using an adverb expressing exaggeration. Biddle et al. (2021) assembled a synthetic dataset called *HyperProbe* [1] including ECFs, qualitative and quantitative hyperboles.

We extract indefinite pronouns that express exaggeration and minimization (e.g., everybody, nobody), quantifiers (e.g., all, none), adjectives using the following POS Tags JJ (adjective), JJR (adjective, comparative) or JJS (adjective, superlative), and adverbs using RB (adverb), RBR (adverb, comparative) or RBS (adverb, superlative). Besides adverb wordlist, another source of descriptive exaggeration terms is distilled from the *HyperProbe* dataset. In Appendix A, we provide in Table 4 detailed resources of the wordlists and dictionaries used to implement our feature extraction methods.

With the help of the *TextBloB* python package [2], we applied POS tagging on text and we extracted **subjectivity** and **polarity**. Persuasion features also include exaggeration, minimization **average proportion** in text, ECF **identifier counts**, boolean **categorical variable** of presence of exaggeration or minimization. (Examples of adverbs used to capture persuasion techniques are listed in the appendix)

Troiano et al. (2018) also considered sentiment analysis and polarity as features to detect exaggeration. Similarly, we extracted subjectivity and emotional intensity as additional persuasion features. Furthermore, subjectivity quantifies the amount of personal opinion and factual information contained in the text. Subjectivity lies within [0,1] where 0 is very objective and 1 is very subjective. The higher subjectivity means that the text contains personal opinion rather than factual information.

3.1.3 Loaded Language

We simply use the "Profanity-filter" package [3] that detects use of profane and offensive terms in text. The function *is_profane* returns a boolean value if this text contains loaded language or not. Profanity-filter only supports English and Russian, we further extend the filter to include other languages and maintain the multilingual system design through implementing dictionaries following the Hunspell [4] format provided by LibreOffice [5].

We use this boolean categorical variable as another persuasion feature in our approach to detect loaded and offensive language. We use the Profanity-filter with deep analysis module which not only matches the exact profane word with custom wordlists but also finds derivative and distorted loaded words in text using the Levenshtein automata (Schulz and Mihov, 2002) initially applied to find fast corrections in string variables.

4 Experimental Results

We ran all classification experiments on a high performing cluster machine with an Intel® Xeon® Gold 6252 (3.70GHz) processor with 24 cores and 48 threads running Linux Red Hat Enterprise Server 8.6 (with Nvidia® Volta V100 GPU for our Pytorch-Lightning [6] implemented system).

On average, it took approximately 31 minutes and 29 seconds to train, validate and predict data across all six languages (More details in Table 5).

In this section, we will provide all experimental results starting by SemEval Task-3 baseline. In accordance with the task organizers' request, all fine-tuning and enhancements made to our system after the deadline have been included in Section 4.3

[1] https://github.com/biddle-r/HyperProbe

[2] https://textblob.readthedocs.io/en/dev/

[3] https://pypi.org/project/profanity-filter/

[4] https://github.com/hunspell/hunspell

[5] https://cgit.freedesktop.org/libreoffice/dictionaries/plain/

[6] https://lightning.ai/docs/pytorch/stable/

4.1 SemEval Baseline

As a baseline for Sub-Task3, SemEval used the Support Vector Machine (SVM) as a base estimator for the multi-output classifier, and it achieved a F1-Micro of 0.16 on the development set and 0.20 on the test set for English. From the provided code, They vectorize the raw input text using unigram and bigram count vectorizer. As preliminary experiments, we tuned other baselines like Random Forest and XGBoost using GridSearch to find a better estimator, but SVM achieved the best results.

4.2 Pre-deadline Results

Our system was developed using the results of XLM-RoBERTA on the development set, so we use it as the foundation model. However, it is easy to switch the first module of our system by simply changing the model name. We tested during development phase different multilingual pre-trained transformers distilled from literature including M-BERT, DistilBERT, XLM-RoBERTA.

In order to be able to make a fair comparison, we further evaluate our initial system on the test set using a range of transformers and the same hyperparameters as before (no tuning was applied at this stage). In Table 1, DistilBERT comes in second place, while XLM-RoBERTA achieves the best results.

4.3 Post-deadline Results

Across all languages, the ReDASPersuasion system has been in constant progress by the help of persuasion features, tuned hyperparameter for model optimization, augmented data and the larger XLM-RoBERTa model. Italian still remains the best with an F1-score of 0.56098 which got us currently ranked second in Italian as well as French.

As illustrated in Figure 2 above, the largest increase percentage in performance on the test set goes to the Polish language, with a 66.4% increase compared to the initial Pre-Deadline model.

We compute F1-Micro percentage increase following Equation (2):

$$Increase = \frac{(PostF1Micro - PreF1Micro)}{|PreF1Micro|} * 100 \quad (2)$$

Detailed information regarding the reasons for this increase in performance will be provided in the subsections that follow.

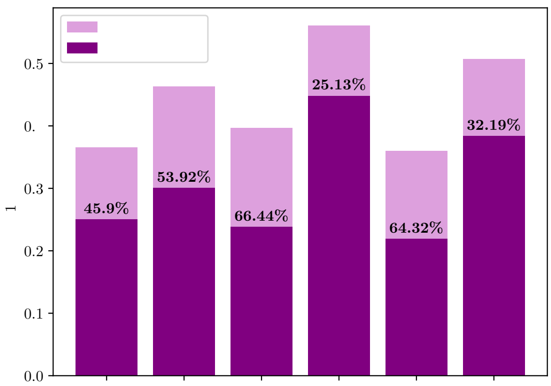

Figure 2: Results on Test Set Before and After Leaderboard Reopening as of 24 April 2023

4.3.1 Fine-tuning and Optimization

For every language, there is a different dataset size, so we fine-tune the hyperparameters following Equation (3) and Equation (4) below:

The number of steps for the warm-up phase:

$$\begin{aligned} Warmup_{Steps} &= \frac{steps_{epoch}}{2} \\ &= \frac{nbr_{train\_samples}}{2 * batch\_size} \end{aligned} \quad (3)$$

The number of total steps:

$$\begin{aligned} Total_{Steps} &= (steps_{epoch} * nbr_{epochs}) \\ &- Warmup_{Steps} \end{aligned} \quad (4)$$

We also changed the XLM-RoBERTa tokenizer and model from the base to the large version, which contains 559 million trainable parameters. This change needs a smaller batch size to run the model on restricted memory. We tuned the hyperparameters in considerations to the model and data size. (see Table 2 reporting the best hyperparameters we obtained for our system optimization)

On the test set, our system works best for Italian language. However, when we use data augmentation and other features, all languages significantly enhance their classification.

4.3.2 Additional Features

Appeal to Time We observed that *Appeal to time* persuasion technique had few samples both on train and dev sets (Table 6 in Appendix B reports that *Appeal to time* has 121 training samples in all 5 languages combined), which impacts the overall

		Classification Results on Test Set					
		En	**Fr**	**Po**	**It**	**Ru**	**Ge**
SubTask 3 Baseline		0.20	0.24	0.18	0.40	0.21	0.32
ReDASPersuasion System	+ *XLM-R*	**0.25**	**0.30**	**0.24**	0.45	**0.22**	**0.38**
	+ *M-BERT*	0.24	0.23	0.07	0.45	0.21	0.33
	+ *DistilBERT*	0.22	0.26	**0.24**	**0.48**	**0.22**	0.36

Table 1: Evaluation of other multilingual transformers within the ReDASPersuasion System on test set. Results in **bold** font represent the best F1-Micro scores

classification of the system. Therefore, we created a new time feature that use adverbs of time with a searching method from the python package "dateparser" [7] that extracts time and date from text.

Repetition We simply use MoreThanSentiments python package (Jiang and Srinivasan, 2023) to extract **redundancy** and **specificity** features in text to catch repeated words and phrases. In fact, redundancy is represented as the percentage of n-grams (n=2) that occur more than once in each document. To detect repetitions in long texts, Cazier and Pfeiffer (2016) used 10-k fillings, showing that the choice of n is directly related to its length. While specificity is the number of specific entity names and quantitative values scaled by the total number of words in a document.

4.3.3 Persuasion Data Augmentation

Static Text Augmenter A common approach for text augmentation is to replace words with their synonyms selected from WordNet (Kober et al., 2020; Li et al., 2022). We use the PPDB database (Ganitkevitch and Callison-Burch, 2014) as source of synonym replacement function. We download paraphrase multilingual packages and use the largest package size XXXL. The PPDB corpora contains a million paraphrases in 16 different languages. These packages include lexical, phrasal and syntactic paraphrasing types.

Dynamic Text Augmenter Rather than using only static word embeddings, we also use contextualized word embeddings to substitute words. In this approach, the length of the sentence is the same, but some words are replaced. Therefore, the augmentation won't impact the max length of the input data, opposite to random insertion that might impact both text length and meaning.

Also, only the most probable tokens with probabilities that add up to $top\_p = 0.95$ or higher are kept for substitution. The *top_p* sampling method works as a control variable for the *diversity* of the textual augmentation.

To obtain both static and dynamic transformations, we use a sequential flow pipeline provided by the python package "nlpaug" [8] to include multiple text augmenters (static and dynamic). We have generated an alteration for each sample to produce double the amount of training and dev examples.

5 Conclusions and Future Work

The model's performance has continually improved since the leaderboard reopened. From recent results, we found that performance improvement is evident as the model size grows (from XLM-R base to large) but also that tuning the model can immensely enhance the model's training and ability to generalize data correctly.

A future direction would be to create an instance of our method to include zero-shot classification tasks using transfer learning. In fact, the transformer has the ability to be trained on zero-shot data, using additional features which can be persuasion attributes in our system. We would like to further explore the persuasive text interpretability per class to better understand the language indicators of persuasion in text.

Acknowledgments

We thank the reviewers for their constructive comments. This research was partly supported by NSF grants 1433817, 1950297 and 2210198, ARO grant W911NF-20-1-0254, and ONR award N00014-19-S-F009. Verma is the founder of Everest Cyber Security and Analytics, Inc.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multi-

[7] https://dateparser.readthedocs.io/en/latest/introduction.html

[8] https://github.com/makcedward/nlpaug

lingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Rhys Biddle, M Rybiński, Qian Li, Cécile Paris, and Guandong Xu. 2021. Harnessing privileged information for hyperbole detection. In *Workshop of the Australasian Language Technology Association*. Australasian Language Technology Association.

Richard A Cazier and Ray J Pfeiffer. 2016. Why are 10-k filings so long? *Accounting Horizons*, 30(1):1–21.

Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 383–389.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *The 9th edition of the Language Resources and Evaluation Conference*, Reykjavik, Iceland. European Language Resources Association.

Jinhang Jiang and Karthik Srinivasan. 2023. Morethansentiments: A text analysis package. *Software Impacts*, 15:100456.

Thomas Kober, Julie Weeds, Lorenzo Bertolini, and David Weir. 2020. Data augmentation for hypernymy detection. *arXiv preprint arXiv:2005.01854*.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90.

Laura Cano Mora. 2009. All or nothing: A semantic analysis of hyperbole. *Revista de Lingüística y Lenguas aplicadas*, 4:25–35.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, SemEval 2023, Toronto, Canada.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Klaus U Schulz and Stoyan Mihov. 2002. Fast string correction with levenshtein automata. *International Journal on Document Analysis and Recognition*, 5:67–85.

Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. 2018. A computational exploration of exaggeration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3296–3304.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

A Implementation Details

Table 2 and Table 3 provide the details about the implemented hyperparameters and libraries, respectively, which are beneficial to help other researchers replicate our experiments. For instance, we fix the random seed to the value 42 for reproducibility.

A.1 Hyperparameters

As seen below in Table 2, some hyperparameters lie within ranges due to the equations described in Equation (3) and Equation (4). Furthermore, each language has a different number of training samples combined with the choice XLM-R version impacts directly these ranges. Larger models require more memory, so we decrease the batch size accordingly and offload the optimizer memory and computation from the GPU to accelerate the training.

Hyperparameters	Range Or Value
Batch Size	8 (XLM-R-large) 32 (XLM-R-base)
Random Seed	42
Learning Rate	4e-05
Warm-up Steps	[12-39]
Total Steps	[1128-4601]
Number of Epochs	30

Table 2: Hyperparameters for System Implementation

A.2 Libraries and Packages

Table 3 shows the implemented python packages used in the *ReDASPersuasion* system discussed above accompanied by the installed version and the purpose in the system's architecture.

Purpose	Python Packages
POS Tagging	*TextBlob* (0.17.1)
Time Appeal	*dateparser* (1.1.7)
Data Augmentation	*nlpaug* (1.1.11)
Redundancy Analysis	*MoreThanSentiments*
Language Detection	*polyglot* (16.7.4)
Language Translation	*translators* (5.5.6)
Language Transformer	*transformers* (4.26.1)
System Implementation	*Pytorch-Lightning* (1.9.2)

Table 3: Python Packages for System Implementation

Additionally, we attach in Table 4 all the dictionaries and word lists used to implements certain linguistic persuasion features to predict exaggeration and loaded language techniques across different languages.

A.3 Time

In Table 5, we report running time to execute the *ReDASPersuasion* system throughout all the six available languages before and after augmentation. We train the model on three GPU devices with distributed data parallel strategy where the model is copied across all GPU devices.

During backpropagation, the resulting gradients across all these copies of the model will be averaged and synchronized. This ensures that each device has the same weights post optimizer step.

B Data Analysis and Statistics

B.1 Persuasion Techniques

Table 6 below represents the total number of persuasion techniques present in the training and development sets. It is important to emphasize the imbalanced nature of these classes, as this has a detrimental effect on classification.

In comparison with confusion matrices on the development sets, the system failed to identify persuasion techniques such as obfuscation, straw man, and whataboutism. They represent only a few examples of both training and development sets.

Future work includes building new features and focusing on qualitative data augmentation to target these particular persuasion techniques that lead to weaker model detection.

B.2 Persuasion Adverbs

Table 7 presents some examples of our exhaustive wordlist that contains a list of adverbs expressing time, doubt, exaggeration degree and minimization.

B.3 Sequence Length

We created a dataset loader module that encodes the textual data using a transformer tokenizer and takes a different max_length which depends on the data set type (training, testing or development sets).

Figure 3 and Figure 4 describe the sequence length distribution of the tokenized samples in train, dev, and test sets. When optimizing the model, we have observed that max_length argument impacts the transformer classification, and it increases time of execution as well.

We would like to explore in the future using models like BigBird (Zaheer et al., 2021) and Longformer (Beltagy et al., 2020) in a cross-lingual environment, to expand the restricted max_length=512 tokens for XLM-R to max_length= 4096 tokens.

This would guarantee the input text tokenization to the maximum token length based on each language. These models are currently also available on the HuggingFace repository [9] [10].

[9] `https://huggingface.co/google/bigbird-roberta-base`

[10] `https://huggingface.co/allenai/longformer-base-4096`

	Wordlists and Dictionaries Sources		
	Loaded Offensive	**Adverbs**	**Dictionaries**
En	profanityfilter badwords list	List of English Adverbs	English Hunspell Dict
Fr	List of French Badwords	French Vocabulary	French Hunspell Dict
Po	List of Polish Obscene and Badwords	Generated List of Adverbs[b]	Polish Hunspell Dict
It	Italian Parole List of Badwords	Generated List of Adverbs[b]	Italian Hunspell Dict
Ru	Russian badwords List	Generated List of Adverbs[b]	Russian Hunspell Dict
Ge	List of German Obscene and Badwords	Generated List of Adverbs[b]	German Hunspell Dict
Multi	Multilingual Badword lists	NA [a]	NA [a]

[a] ***NA** : Not Available.* [b] *Translated from other sources (En + Fr) using Google Translation.*

Table 4: Dictionaries and Word Lists Resources

	Total Running Time (min:sec)	
	Before DA [a]	*After DA* [a]
En	(58:12)	(75:28)
Fr	(32:47)	(55:43)
Po	(10:55)	(29:50)
It	(30:03)	(46:24)
Ru	(25:09)	(37:28)
Ge	(31:50)	(52:16)
Avg[b]	(31:29)	(49:31)

[a] ***DA** : Data Augmentation.* [b] ***Avg** : Average Time.*

Table 5: Total Time for System Execution Per Language using 3 GPU devices

		Number of Persuasion Techniques					
		En	Fr	Ge	It	Pl	Ru
Appeal to Authority	Train	154	76	225	70	41	10
	Dev	28	40	36	24	40	2
Appeal to Fear Prejudice	Train	310	210	182	285	108	54
	Dev	137	62	45	85	36	13
Appeal to Hypocrisy	Train	40	134	136	82	162	103
	Dev	8	37	56	27	76	17
Appeal to Popularity	Train	15	82	63	37	30	8
	Dev	34	17	17	18	22	2
Appeal to Time	Train	NA‡	41	11	27	14	28
	Dev	NA‡	14	6	16	5	1
Appeal to Values	Train	NA‡	100	73	131	101	48
	Dev	NA‡	44	36	55	50	8
Causal Oversimplification	Train	213	125	33	50	12	39
	Dev	24	44	20	12	5	6
Consequential Oversimplification	Train	NA‡	112	35	29	24	70
	Dev	NA‡	53	12	9	8	13
Conversation Killer	Train	91	170	121	178	50	88
	Dev	25	52	31	69	40	24
Doubt	Train	518	327	288	882	295	509
	Dev	187	95	93	287	96	107
Exaggeration Minimisation	Train	466	258	157	143	111	131
	Dev	115	74	43	48	40	27
False Dilemma No choice	Train	122	73	41	61	12	28
	Dev	63	29	5	16	8	11
Flag Waving	Train	287	37	65	35	68	42
	Dev	96	10	18	12	28	10
Guilt by Association	Train	59	130	122	53	94	24
	Dev	4	29	23	22	30	7
Loaded Language	Train	1809	944	242	903	310	641
	Dev	483	250	77	296	93	150
Name Calling Labeling	Train	979	428	734	566	475	253
	Dev	250	116	240	181	111	43
Obfuscation Vagueness Confusion	Train	18	113	62	21	36	19
	Dev	13	36	22	4	11	10
Questioning Reputation	Train	NA‡	348	310	383	164	303
	Dev	NA‡	87	80	122	57	94
Red Herring	Train	44	55	30	23	12	2
	Dev	19	9	4	4	7	1
Repetition	Train	544	92	8	22	13	69
	Dev	141	21	4	15	10	20
Slogans	Train	153	149	87	54	36	72
	Dev	28	27	39	20	7	11
Straw Man	Train	15	135	15	51	15	21
	Dev	9	23	2	15	3	9
Whataboutism	Train	16	62	13	8	8	7
	Dev	2	12	13	1	3	4

‡ ***NA** : Persuasion Not Available in English Set (Only 19 Techniques).*

Table 6: Total Number of Labeled Persuasion Technique per Language in Sub-Task 3

Adverb Type	Examples
Time	*[daily, constantly, today, now, before]*
Doubt	*[certainly, possibly, honestly, truly]*
Minimization	*[few, simply, somewhat, least, little]*
Exaggeration	*[very, highly, really, terribly, extremely]*

Table 7: Examples of Adverbs for persuasion techniques

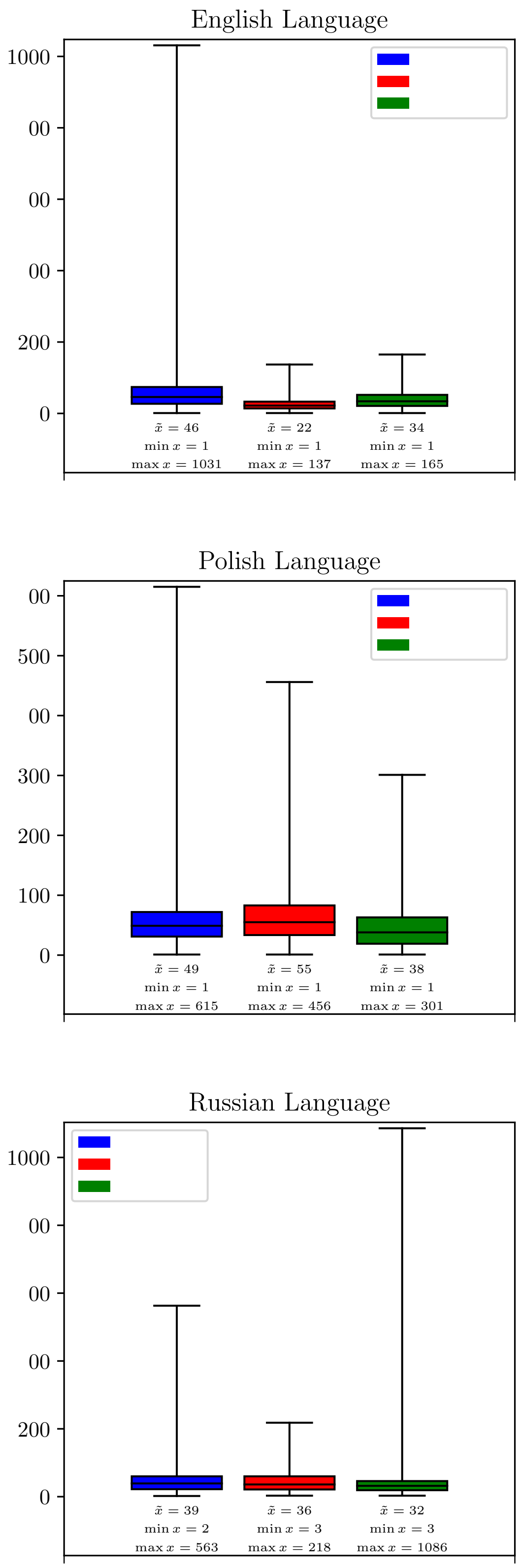

Figure 3: Sequence Token Length of SemEval 2023 Sub-task3 data sets (English, Polish, and Russian). $\tilde{x}$ denotes the median value of text length.

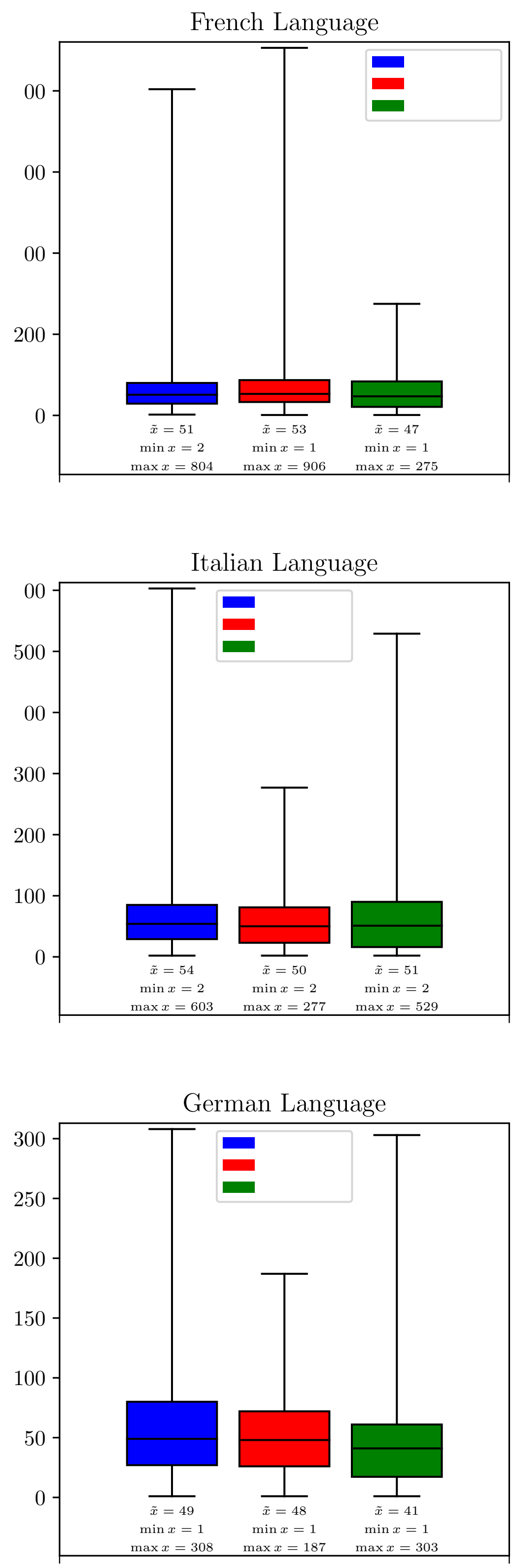

Figure 4: Sequence Token Length of SemEval 2023 Sub-task3 data sets (French, Italian, and German). $\tilde{x}$ denotes the median value of text length.