Team Oulu at SemEval-2020 Task 12: Multilingual Identification of Offensive Language, Type and Target of Twitter Post Using Translated Datasets

Md. Saroar Jahan
University of Oulu,
Faculty of Information Tech., CMVS
PO Box 4500, Oulu 90014 FINLAND
mjahan18@student.oulu.fi

Mourad Oussalah
University of Oulu,
Faculty of Information Tech., CMVS
PO Box 4500, Oulu 90014 FINLAND
mourad.oussalah@oulu.fi

Abstract

With the proliferation of social media platforms, anonymous discussions together with easy online access, reports on offensive content have caused serious concern to both authorities and research communities. Although there is extensive research in identifying textual offensive language from online content, the dynamic discourse of social media content, as well as the emergence of new forms of offensive language, especially in a multilingual setting, calls for future research in the issue. In this work, we tackled Task A, B, and C of Offensive Language Challenge at SemEval2020. We handled offensive language in five languages: English, Greek, Danish, Arabic, and Turkish. Specifically, we pre-processed all provided datasets and developed an appropriate strategy to handle Tasks (A, B, & C) for identifying the presence/absence, type and the target of offensive language in social media. For this purpose, we used OLID2019, OLID2020 datasets, and generated new datasets, which we made publicly available. We used the provided unsupervised machine learning implementation for automated annotated datasets and the online Google translation tools to create new datasets as well. We discussed the limitations and the success of our machine learning-based approach for all the five different languages. Our results for identifying offensive posts (Task A) yielded satisfactory accuracy of 0.92 for English, 0.81 for Danish, 0.84 for Turkish, 0.85 for Greek, and 0.89 for Arabic. For the type detection (Task B), the results are significantly higher (.87 accuracy) compared to target detection (Task C), which yields .81 accuracy. Moreover, after using automated Google translation, the overall efficiency improved by 2% for Greek, Turkish, and Danish.

1 Introduction

The emergence of Web 2.0 platform that enabled user-generated content and participatory culture has witnessed the proliferation of online hate speech at an unprecedented level, increasing the likelihood of a random people of any age group to be subject to online harassment and abuse through some internet forum, message board or social network platform. On the other hand, offensive language fosters discrimination against particular categories on the basis of gender, ethnicity, or race (Nockleby, 2000), which violates the Universal Declaration of Human Rights. This often leads to a profound negative impact on the society as a whole, especially for youth and vulnerable groups, as well as on the policy-makers and business entities. To overcome this issue, many internet companies put forward standards and generic guidelines that users must adhere to when publishing content on such platforms; at the same time, they employ manual annotators to identify offensive language and remove the post (s) accordingly. Nevertheless, such manual effort is inevitably costly, non-scalable, and non-sustainable, which motivates the importance of automatic offensive language identification based approach. Prior work has studied offensive language detection in Twitter (Burnap and Williams, 2015; Wiegand et al., 2018; Foong and Oussalah, 2017), Wikipedia comments and Facebook posts (Kumar et al., 2018), and Fromspring posts (Reynolds et al., 2011). Several academic events, shared task competition organized in conjunction with high impact data mining, information retrieval and computational linguistics conferences (e.g., Workshop series on Abusive Languages, Automatic Misogyny Identification, Authorship Aggressiveness Analysis, Identification of Offensive Language at GermEval, Hate Speech Detection Task at Evalita, various related SemEval tasks,

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1628–1637
Barcelona, Spain (Online), December 12, 2020.

etc.). Some of these tasks involved more than one language. In this respect, one of the most commonly employed methodologies is to train systems that can automatically identify offensive content, which will then trigger action to remove such content without any human moderation. Past research has also examined various characteristics of offensive language such as the cyber aggression (Kumar et al., 2018), abusive language (Nobata et al., 2016; Mubarak et al., 2017), hate speech (Foong and Oussalah, 2017; Abderrouaf and Oussalah, 2019), Racism (Kwok and Wang, 2013) and offensive language (Wiegand et al., 2018). Nevertheless, automatic identification of offensive language is still challenging, especially given the continuous evolution and variability of offensive language discourse and characteristics together with the inherent limitation of natural language processing based approach. Besides, the use of multiple languages in a single post adds extra difficulty to such a task. In this context, (Zhang et al., 2019) used an ensemble classifier based approach to identify hate-speech against immigrants and women in Spanish and English Twitter datasets. (Zampieri et al., 2019a) as part of SemEval2019 Task, addressed the issue of categorizing offensive language in social media. As part of our contribution to the Offensive Language Challenge at SemEval2020, our work presents a three-level annotation schema for abusive language detection, categorization, and target detection. First, we created an Offensive Language Identification Dataset (OLID) by using unsupervised learning and thresholding-like technique. All the five provided languages (English, Turkish, Greek, Danish, and Arabic) were employed for the offensive post detection task. Moreover, we have used Google online tools for creating automated translation of datasets for Turkish, Greek, Danish, and Arabic. We performed experiments using different machine learning models and features for each level of the annotation, which enabled us to compare the performance of the machine learning automated annotated datasets and Google translated dataset. The outcome is then used to trigger subsequent data analysis tasks of multilingual detection. The paper is structured as follows. In Section 2, our dataset annotation schema and description of datasets are detailed, section 3 details our methodology, including the machine learning model and the associated feature engineering. In Section 4, the obtained results are detailed and discussed. Finally, conclusive statements and potential future work are drawn in the Conclusion section.

2 Datasets

2.1 Datasets Annotation

While working with OLID2019[1] (Zampieri et al., 2019b) and OLID2020[2] (Zampieri et al., 2020) datasets, we have followed OLID guidelines in labelling the datasets and split them into three subtasks to distinguish i) whether the language is offensive or not (Task A); ii) type of the language (Task B) and; iii) target entity (Task C). Each task and label has been described in detail below, and examples of datasets are shown in Table 1.

Example of Tweet message	**Task A**	**Task B**	**Task C**
She should ask a few native Americans what their take on this is	OFF	UNT	-
Go home you're drunk!!! MAGA Trump2020 ... URL	OFF	UNT	IND
Someone should've Taken this piece of shit to a volcano	OFF	UNT	-
Obama wanted liberals & illegals to move into red states	NOT	-	-
Liberals are all Kookoo	OFF	TIN	OTH

Table 1: Example tweets and annotation from English datasets.

Task A (Multilingual): Offensive language Detection Task-A for English, Greek, Danish, Arabic, and Turkish languages, and Google English translation of all these languages:

i) Not Offensive (NOT): Posts that do not contain any offensive word or profanity.

[1] https://sites.google.com/site/offensevalsharedtask/offenseval2019 (accessed June 30, 2020)

[2] https://sites.google.com/site/offensevalsharedtask/home (accessed June 30, 2020)

ii) Offensive (OFF): A post containing offensive words or targets an (offense direct or indirect). We label a post as offensive (OFF) if it contains any form of improper language (profanity) or is offensive, which can be expressed directly or indirectly. This includes insulting posts, sexual nature posts, threats, and other kinds of posts containing profane language or swear words.

Task B: Categorization of Offensive Language type Task-B for English language:

i) Targeted Insult (TIN): Posts containing insult/threat targeting an individual, a community etc.

ii) Untargeted (UNT): Posts containing non-targeted profanity or swear words. Posts with general profanity are not targeted, but contain non-acceptable language.

Task C: Offensive Language Target Identification Task-C for English language:

i) Individual (IND): Posts targeting an individual. This can be a renowned person, a named individual (cyberbullying) or an unnamed participant in the conversation.

ii) Group (GRP): Posts targeting a group of people considered as a unity due to the same ethnicity, gender or sexual orientation, political affiliation, religious belief, or other common characteristics, which is commonly understood as hate speech.

iii) Other (OTH): The target of these offensive posts does not belong to any of the previous two categories (e.g., an organization, a situation, an event, or an issue).

Datasets Name	**Size**
English Task A, B and C, Old Datasets OLID2019	13k
English Task A; unsupervised learning based annotated from OLID2020	16.5k
English Task B; unsupervised learning based annotated from OLID2020	8.5k
English Task C; unsupervised learning based annotated from OLID2020	6.3k
Greek OLID2020	8.5k
Greek OLID2020 to English translated	8.5k
Turkish OLID2020	31k
Turkish OLID2020 to English translated	31k
Arabic OLID2020	7k
Arabic OLID2020 to English translated	7k
Danish OLID2020	2.9k
Danish OLID2020 to English translated	2.9k

Table 2: Datasets name and size.

2.2 Datasets Descriptions

To train our models and compare our results, we have used four different datasets: OLID-2019, OLID-2020, automated annotated datasets, and automated Google translated (English, Turkish, Greek, Danish and Arabic) datasets (Table 2 shows different dataset's names and sizes).

i) **OLID-2019 and OLID-2020:** OLID-2019 (Zampieri et al., 2019a) dataset was already annotated by SemEval-2019 competition for English subtask A, B and C. However, OLID-2020 dataset is the new dataset which was prepared for SemEval-2020 completion. These new datasets were annotated and are ready to use for Arabic (Mubarak et al., 2020), Danish (Sigurbergsson and Derczynski, 2020), Greek (Pitenis et al., 2020), and Turkish (Çöltekin, 2020) languages. However, English (Rosenthal et al., 2020), datasets were not fully annotated for Task A, B, and C; instead, they provided scores of Average (AVG) and Standard Deviation (STD). The scores were confidence measures produced by unsupervised learning methods using OLID-2019 datasets. More specifically:

- AVG: are the averages of the confidences predicted by several supervised models for a specific instance that belongs to the corresponding class.

- STD: are the standard deviations from 'AVG' for a particular instance.

ii) **Automatically annotated English datasets by using unsupervised learning method:** To generate new training datasets and to label them automatically, we have to map the scores to the labels through some thresholding. Empirical analysis through trials and incremental error-correcting procedures have been employed to set up threshold values by manually checking the quality of the first 200 posts after each incremental choice of threshold. We have set thresholds for Task A: $0.09 > AVG = NOT, 0.95 <$

$AVG = OFF$, Task B: $0.24 < AVG < 0.6 = TIN, 0.6 < AVG = UNT$, and Task C: $0.5 < AVG < 0.7 = OTH, 0.7 < AVG < 0.8 = IND, 0.8 < AVG = GRP$. Finally, we have saved the newly generated dataset as CSV files. We have made our newly generated dataset publicly available.

Google translated datasets: An online Google translation tool[3] was used to translate Turkish, Greek, Danish, and Arabic words of the OLID2020 dataset. Before the translation, we preprocessed the datasets so that Google translation API would work accurately. In addition, we separated annotations from the posts so that we could upload only posts containing short files to the online tools for the maximum translation performance. When the translation was completed, we merged the files containing annotations and then post the translated file together to make a complete translated dataset (Figure 1).

For English language Task A, B, and C, we have used both OLID2019 old and newly generated datasets by using unsupervised machine learning techniques employed as part of the SemEval 2020 dataset preparation. However, for Danish, Turkish, Greek, and Arabic languages, we have used OLID2020 dataset, which are manually annotated and provided by OLID, and Google translated datasets for all these five languages.

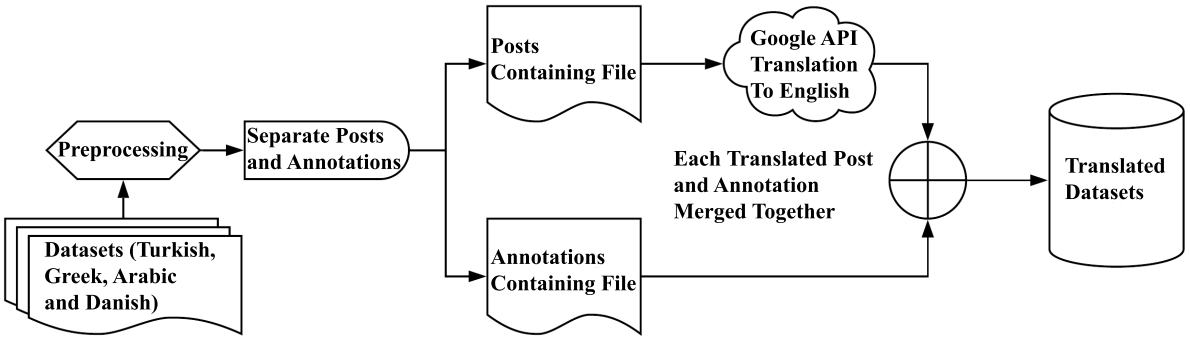

Figure 1: General synoptic of translation process.

3 Methodology

3.1 Preprocessing

All datasets (English, Turkish, Danish, Arabic, Greek, and translated datasets) were preprocessed using standard NLP preprocessing tools. However, some preprocessing job cannot be processed for all the five languages. For example, special character removal was not possible for Greek language. Similarly, uppercase to lowercase conversion was only doable for English language. Besides, we could not convert abbreviated and short slangs to the original word except for the English language due to fewer resources and limited parser capabilities of those languages. For stop-word removal, we have made a list of stop-words for each language and applied it accordingly. In summary, the preprocessing is performed through the following:

i) Converted words to lowercase. (For English language only).

ii) Filtered out stop-words (for example, a, an, the, etc.). (For all languages).

iii) Removed single characters (except those characters could have abbreviated meaning ex. 'F' character), excluding emoji's.

iv) Removed URLs with a space. (For all languages).

v) Removed Twitter hashtag (#) and user (@user). (For all languages).

vi) Abbreviated words and short forms of social network slangs are replaced with original words,for example, fag to faggot. (For English language only).

vii) Unidentified characters, symbols removed. (Exception for Greek language).

viii) Tab token and multiple spacing has been removed. (For all languages).

[3] `https://www.onlinedoctranslator.com/` (accessed June 30, 2020)

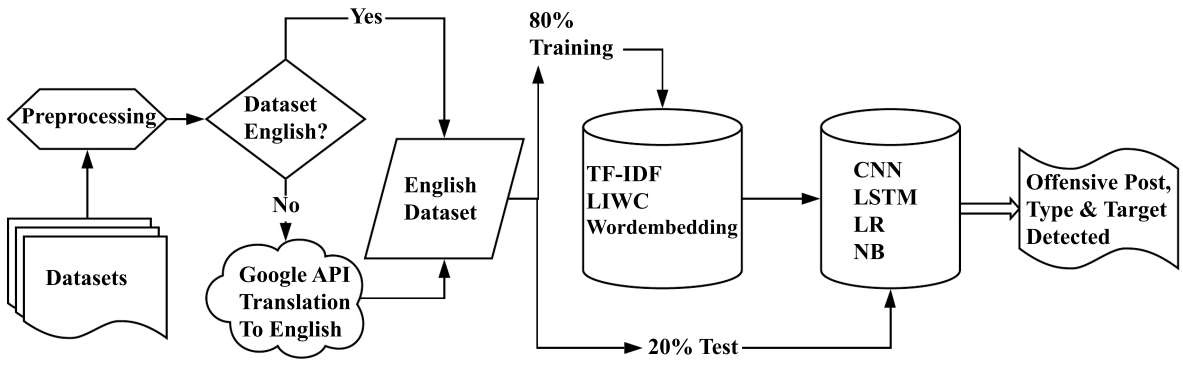

Figure 2: A general synoptic of the system.

3.2 Feature Engineering

A set of features have been employed and evaluated for the purpose of Task A, B, and C. We have used five (5) types of features: Word-level, N-Gram word-level (for N=2, 3), N-Gram Character-level (for N=3, 4), word embedding and sentiment analysis. Word level TF-IDF feature assigns a score to every term in documents, while word-level N-gram feature applies TF-IDF scoring to all 2-grams and 3-grams tokens extracted from the whole corpus dataset. Character Level TF-IDF provides a matrix representation of TF-IDF scores of character-level n-grams in the corpus. We restricted to 5000 features for each type in order to avoid the computational burden.

Word Embeddings Features: A word embedding is a procedure of demonstrating words and documents with a dense vector representation. The position of a word inside the vector space is learned from the text and is constructed on the words that surround the word once it is used. We can train the word embedding using the input corpus itself; however, in this paper, we have used the pre-trained word embedding, namely FastText[4] (Mikolov et al., 2017). First, we created a tokenizer and converted the text of train data to a sequence of tokens and pad them to ensure equal length vectors with a max length of 70. Then we used an embedding matrix with 300 elements each.

Sentiment Analysis: Linguistic Inquiry and Word Count (LIWC) features provide more than 90 features, including word counts, psychological process, cognitive process categories, summary variables, etc. In this work, we restricted the LIWC features to only categories that convey sentiment: Positive and Negative emotion. We have used a python library, namely 'Empath[5] LIWC (Linguistic Inquiry and Word Count)'. The LIWC program includes the main text analysis module, along with a group of built-in dictionaries. Once the processing module has read and accounted for all words in a given text, it calculates the percentage of total words that match each of the dictionary categories. In other words, for a given input post message, we compare each work of the message with the LIWC build in dictionary and records the matching categories. For instance, if 15 negative emotion words are found, this number is converted to a 15% negative emotion.

3.3 Classifier Architecture

Initially, we employed a random split of the original dataset into 80% for training and 20% for testing and validation, ensuring the same proportion of dataset for all kinds of datasets in order to ensure a balanced training. Four types of classifiers were implemented for English datasets: Convolution Neural Network (CNN), the recurrent neural network LSTM models, Linear regression, and Naive Bayes. The classifier that outperformed other classifiers in 'English' language experiment has been used for Greek, Danish, Turkish and Arabic datasets rather than using all classifiers. Table 3 shows datasets and classifier names that have been used.

[4] `https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki-news-300d-1M.vec.zip` (accessed Dec 30, 2018)

[5] `https://github.com/Ejhfast/empath-client` (accessed June 30, 2020)

Datasets Name	Classifier Used
English Task (A, B, and C)	CNN, LSTM, Linear regression, Naive Bayes
Danish Task A	Linear regression, CNN
Turkish Task A	Linear regression, CNN
Arabic Task A	Linear regression, CNN
Greek Task A	Linear regression, CNN

Table 3: Task and Classifier names that presented in the result sections.

We adopted (Kim, 2014) CNN, architecture, where the input layer is represented as a concatenation of the words forming the post (up to 70 words), except that each word is now represented by its FastText embedding representation with 300 embedding vector. A convolution 1D operation with a kernel size 3 was used together with a max-over-time pooling operation over the feature map with a layer dense 50. Dropout on the penultimate layer with a constraint on l2-norms of the weight vector was used for regularization. Similarly, the LSTM scheme is similar to the word embedding representation as in our CNN model. However, we have followed (van Aken et al., 2018) where LSTM layers have 128 units, followed by a dropout of 10%. On the other hand, two baselines algorithms that use Linear Classifier (Logistic Regression) and Naive Bayes Classifier were considered for comparison purposes. The details of the implementation are reported in our GitHub page of this project with datasets and codes[6] . The various features were examined by each classifier in order to test its accuracy and robustness.

4 Results

Preprocessing Type	English	Turkish	Greek	Danish	Arabic
All	0.891	0.84	0.85	0.92	0.885
URL, Special Ch.	0.891	0.84	-	0.91	0.88
USERNAME (user)	0.884	0.83	0.838	0.913	0.878
Newline, Tab Token	0.891	0.84	0.84	0.913	0.89
Abbreviated word	0.890	-	-	-	-
Emoji	0.88	0.827	0.84	0.91	0.88
No Preprocessing	0.873	0.822	0.83	0.90	0.87

Table 4: Accuracy scores changes in preprocessing for all language using LR Model and TF-IDF word level feature.

4.1 Preprocessing Outcomes

The results of different types of preprocessing for all languages are summarized in Table 4 that records the classifier accuracy for offensive language detection (Task-A) using LR classifier with TF-IDF word level feature. The latter were only employed for illustration purpose to comprehend the effects of the various preprocessing schemes. The results highlighted in Table 4 for preprocessing task indicate the following:

For English language, the use of uppercase to lowercase conversion in the preprocessing stage does not affect much the overall result.

Stop-word and emoji removal works for all languages and increases by 1% the accuracy. However, Newline + Tab Token, and URL + Special Characters removal work well for all languages and improved almost 2% in performance accuracy each.

Finally, when applying all preprocessing possibilities -all together- almost the five languages show improvement with a classifier accuracy close to 2.5%. This indicates that language preprocessing techniques for NLP impact the performance of offensive language detection for all languages. This also motivates the use of the –all together preprocessing in subsequent tasks.

[6] `https://github.com/saroarjahan/SemEval2020` (accessed July 24, 2020)

4.2 Offensive Language Detection Task A

The performance on identifying between offensive (OFF) and non-offensive (NOT) posts is reported in Table 5 , 6 and 7. Table-5 represents the results for English language with four (4) different models and three (3) different features. On the other hand, Table-6 highlights the results for other four (4) languages with only the best classifier and feature recorded for the English language. Similarly, Table-7 summarizes the results for multilingual while using the automated Google translated to English datasets. Commenting on the results in Table 5, 6 and 7 for offensive and non-offensive post detection, one shall notice the following:

Classifier	OLID2019 Dataset		Artificially Annotated Dataset		Both Datasets	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
NB + WordLevel TF-IDF	0.87	0.81	0.92	0.88	0.89	0.85
NB + WordLevel TF-IDF+LIWC	0.874	0.815	0.923	0.884	0.895	0.856
NB + N-Gram TF-IDF	0.87	0.82	0.92	0.87	0.89	0.85
NB + N-Gram TF-IDF+LIWC	0.874	0.823	0.924	0.875	0.894	0.854
NB + CharLevel TF-IDF	0.87	0.81	0.92	0.88	0.90	0.85
NB + CharLevel TF-IDF+LIWC	0.876	0.813	0.925	0.884	0.903	0.854
LR + WordLevel TF-IDF	0.89	0.86	0.93	0.91	0.92	0.90
LR + WordLevel TF-IDF+LIWC	0.896	0.865	0.936	0.916	0.926	0.905
LR + N-Gram TF-IDF	0.87	0.81	0.92	0.88	0.89	0.85
LR + N-Gram TF-IDF+LIWC	0.875	0.815	0.924	0.884	0.893	0.853
LR + Char Level TF-IDF	0.89	0.87	0.94	0.92	0.92	0.90
LR + Char Level TF-IDF+LIWC	0.898	0.876	**0.948**	**0.927**	0.926	**0.905**
CNN + WordEmbedding	**0.903**	**0.885**	0.943	0.922	**0.932**	0.90
LSTM + WordEmbeddings	0.90	0.88	0.94	0.92	0.92	0.90

Table 5: Accuracy and F1 scores for Offensive Language detection Task A, English dataset (best in bold).

Classifier	Turkish		Danish		Greek		Arabic	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
NB + WordLevel TF-IDF	0.84	0.75	0.81	0.74	0.85	0.83	0.89	0.87
LR+ CharLevel TF-IDF	0.83	0.74	0.80	0.73	0.85	0.82	0.88	0.86

Table 6: Accuracy and F1 scores for Offensive Language detection Task A, Multilingual (best in bold).

In Table-5 for OLID old datasets, automated annotated datasets and both datasets together yield almost similar Accuracy and F1 scores; however, an artificially annotated dataset developed by the SemEval2020 unsupervised learning method produced 2-4% better performance compared to old datasets. In addition,

Classifier	Turkish to English		Danish to English		Greek to English		Arabic to English	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
LR+Word Level TF-IDF	0.86	0.75	0.82	0.76	0.87	0.86	0.86	0.83
LR+Word Level TFIDF+LIWC	0.87	0.77	0.83	0.75.5	0.88	0.86	0.86	0.83
LR + Char Level Vector	0.84	0.74	0.81	0.74	0.87	0.85	0.85	0.82
LR+ Char Level Vect+LIWC	0.85	0.75	0.81	.74	0.87	0.86	0.86	0.82
CNN +Word Emb.	0.85	0.77	0.82	0.75	0.88	0.86	0.86	0.83

Table 7: Accuracy and F1 scores for Offensive Language detection Task A, Multilingual translated to English datasets (best in bold).

when old datasets and new datasets were jointly combined, this yielded a better performance compared to old datasets alone. This indicates that artificially annotated datasets can be useful in many NLP based applications, including offensive post detection.

Although the performance of all models is at a pretty acceptable level, the CNN model slightly outperforms LR and LSTM models, achieving a maximum accuracy and F1 score of 0.94. In contrast, Naive Bayes presents a low performance compared to others.

In table-6 & 7, the use of TF-IDF + LIWC provides as expected the best result and improved by 1% in terms of accuracy and F1-measure compared to individual features.

In the multilingual result of Table 6, LR with Word Level TF-IDF model outperforms all other models. For Turkish, Danish, Greek and Arabic languages, the accuracy and the F1 scores are (0.84, 0.75), (0.81, 0.74), (0.85,0.83) and (0.89, 0.87), respectively. Among the five languages, the offensive-speech detection task is more successful for the English language, possibly because of the quality of parser and subsequent NLP tools. For identifying offensive posts (Task A), all the five languages yielded quite similar performance in terms of accuracy and F1 scores, which indicates common difficulty trend in recognizing offensive languages accross languages.

Table-7 and 8 show that both the original (non translated) and the Google translated datasets' scores yielded close performance in terms of accuracy and F1 scores. However, Google translated datasets show a performance improvement of 2% in accuracy for Turkish to English and Danish to English translations. In contrast, Greek to English translation improved the efficiency by approximately 3% compared to non-translated datasets. Among all these four languages, only Arabic to English translation shows a decrease in performance by 2% as compared to non-translated datasets. Clearly, this result indicates that translation of a language to English could be useful for most of the languages, and sometimes it may yield better accuracy due to a better available of resources and efficient NLP tools for the English language.

4.3 Categorization (Task B) and Target Identification (Task C)

Classifier	OLID2019 Dataset		Artificially Annotated Dataset		Both Datasets	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
NB + WordLevel TF-IDF	0.84	0.83	0.87	0.85	0.85	0.84

Table 8: Accuracy & F1 scores for type detection Task B, English dataset.

Classifier	OLID2019 Dataset		Artificially Annotated Dataset		Both Datasets	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
NB + WordLevel TF-IDF	0.79	0.78	0.81	0.79	0.80	0.78

Table 9: Accuracy & F1 scores for target detection Task C, English dataset.

In the results shown in Table 8, the models were trained to discriminate between targeted insults and threats (TIN) and untargeted (UNT) offenses.

In Table 9, the models were trained to discriminate between targeted individuals (IND) and groups (GRP) and or others (OTH).

In tables 8 and 9, OLID old datasets, new datasets, and both datasets yield almost similar accuracy and F1 Scores for Task B & C.

We observe that 'type' detection outperforms 'target' detection by almost 6%. One of the reasons for this result could be the fact that Task B has only 2 features, while Task C has 3 features to train the model. In addition, Task C data size was smaller compared to Task B. Besides, Task C features were closely related to each other, leaving less room for diversification. For instance, the difference between IND and GRP was very crucial for the differentiation purposes.

The new dataset generated by OLID-2020 through the unsupervised learning method yielded 3% performance increase compared to old datasets. In addition, when old datasets and new datasets were applied jointly, it outperformed the performance of old dataset by a 1% margin. This indicates that artificially annotated datasets can be useful for the type and target detection purpose as well.

5 Conclusion and Future Work

This paper summarizes our participation to SemEval2020 in Tasks A, B, and C. The methodology advocates a two-stage strategy with an initial preprocessing that involves both automated annotated using unsupervised learning and Google translation dataset and machine learning-based classification using CNN and FastText embedding. In addition, we have compared the results with the manually annotated datasets and artificially annotated datasets. Our result shows that artificially annotated dataset using unsupervised learning yields 2-4% high performance, and in all cases, CNN and LSTM outperform baseline classifiers. Moreover, our work presented a sound comparison of the results across five distinct languages: English, Greek, Turkish, Arabic, and Danish. We observed that the developed approach could overcome the language barrier, maintaining high-performance levels across the five languages. The analysis has also shown that the performance can be slightly improved by tuning the preprocessing stage. Moreover, we have observed that Google translation of dataset yielded 2-3% higher performance for Greek, Turkish, and Danish, except for the Arabic language. This result promotes the viability of use of automated translated datasets in NLP projects. In addition, we found 'target' detection was hardest compared to 'offensive post' and 'type' detection. In future work, we would like to experiment with broader category detection and apply more specialized CNN classifiers that enable wordsense disambiguation.

Acknowledgements

This work is partly supported by European project YoungRes (#823701), which is gratefully acknowledged.

References

Cheniki Abderrouaf and Mourad Oussalah. 2019. On online hate speech detection. effects of negated data construction. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5595–5602. IEEE.

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.

Çağrı Çöltekin. 2020. A Corpus of Turkish Offensive Language on Social Media. In *Proceedings of the 12th International Conference on Language Resources and Evaluation*, pages 6174–6184. ELRA.

Yee Jang Foong and Mourad Oussalah. 2017. Cyberbullying system detection and analysis. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 40–46. IEEE.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the first workshop on abusive language online*, pages 52–56.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on twitter: Analysis and experiments. *arXiv preprint arXiv:2004.02192*.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

John T Nockleby. 2000. Hate speech. *Encyclopedia of the American constitution*, 3(2):1277–1279.

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive Language Identification in Greek. In *Proceedings of the 12th Language Resources and Evaluation Conference*. ELRA.

Kelly Reynolds, April Kontostathis, and Lynne Edwards. 2011. Using machine learning to detect cyberbullying. In *2011 10th International Conference on Machine learning and applications and workshops*, volume 2, pages 241–244. IEEE.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. In *arxiv*.

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive Language and Hate Speech Detection for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*. ELRA.

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

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

M Zampieri, P Nakov, S Rosenthal, P Atanasova, G Karadzhov, H Mubarak, L Derczynski, and Z Pitenis. 2020. Cöltekin, c.(2020). semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *Proceedings of SemEval*.

Mike Zhang, Roy David, Leon Graumans, and Gerben Timmerman. 2019. Grunn2019 at semeval-2019 task 5: Shared task on multilingual detection of hate. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 391–395.