WEAK SUPERVISION FOR NAMED ENTITY RECOGNITION AND OFFENSIVE LANGUAGE IDENTIFICATION

KIERON SEVEN LEE JUN WEI

2022

Abstract

This project examines the feasibility of using weak supervision to train state-of-the-art language models. Machine learning models were fine-tuned to perform Natural Language Processing (NLP) tasks such as Named Entity Recognition (NER) and Offensive Language Identification (OLI). English comments from online forums popular in Singapore like Reddit and HardwareZone were used as training data. Transfer learning, in the form of fine-tuning, was performed on transformer models from spaCy using labels generated through weak supervision by skweak and Hugging Face Transformers libraries. For both tasks, these weak supervision models' accuracy were within 5% of the same transformer models fine-tuned on ground-truth labels. A web application was also created to demonstrate the models' capabilities on unseen text.

Acknowledgments

I would like to express my deepest gratitude to the following people for making this project a success.

- Defence Science and Technology Agency (DSTA), for offering me the internship opportunity.
- My supervisors: Mr Wang Shunde, Mr Ian Lee and Mr Martin Chan from Information PC, DSTA, for their patience and expertise in guiding me through the various challenges I faced. They have made my internship experience a thoroughly enjoyable one.
- All my colleagues and other parties not mentioned here for their support during this internship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, the introduction of deep learning transformer models has been an inflexion point for the domain of form Natural Language Processing (NLP). These state-of-the-art models utilise a self-attention mechanism that enables transfer learning, compared to existing neural networks like Long Short-Term Memory [1]. Hence, a transformer model like Bidirectional Encoder Representations from Transformers (BERT) can be pre-trained on large datasets before the user fine-tunes the model for a specific NLP task. However, the efficiency of these models is bottlenecked by the availability of high-quality labelled training data needed for transfer learning [2]. This limitation could be due to the usage of classified data or subject matter experts lacking time to manually label data [3].

1.2 Weak Supervision

Weak supervision is one solution that overcomes the necessity of manually labelled training data for fine-tuning. By leveraging imperfect, noisier inputs such as heuristics, patterns and external knowledge bases, supervision signals can provide labels for large amounts of training data [3]. This project explores weak supervision for NLP tasks such as Named Entity Recognition (NER) and Offensive Language Identification (OLI). For example, NER is typically performed by recognising entities through tokenizing and parsing a sentence before categorising these entities using a corpus [4]. The absence of a local corpus for this task can be overcome by aggregating entity labels from weakly supervised heuristics and resolving conflicting labels.

Chapter 2

Data

2.1 Data Collection

Comments from popular local forums were used to train and evaluate the weak supervision models. This project obtained comments made on user-submitted posts from the "Singapore" subreddit (r/Singapore) on Reddit [5] and the "Eat-Drink-Man-Woman" forum (EDMW) on HardwareZone [6] across a period of two weeks in April 2022. The message content of the comment, username of the commenter and the Uniform Resource Locator (URL) to the post were extracted from both online forums. Comments in the Offensive Language Identification Dataset (OLID) were also used as a benchmark against the models' performance on the OLI task. The complete OLID dataset with training and testing labels was downloaded from the OffensEval website [7].

2.2 Web Scraping

2.2.1 r/Singapore

This project used custom Python scripts to automate the process of obtaining comments from both forums. A script using the Python Reddit API Wrapper (PRAW) library [8] scraped comments from posts in r/Singapore through the Reddit API. The highest upvoted posts of different periods: day, week, month, year and all-time, were chosen as these posts tend to have higher visibility and contain more comments. Around 18,000 unique comments from r/Singapore's user-submitted posts about different topics were obtained. Comments that were shorter than ten words were filtered out as they had insufficient named entities for the NER task. Automated programmes or bots on Reddit had their comments removed as well. Bots were usually identified by the phrase 'bot' in their Reddit username, or their comment would contain a repeated message.

2.2.2 EDMW

This project deployed another script using the Python Beautiful Soup library [9] to extract comments from EDMW as HardwareZone does not have an API. The script accessed forum threads from the first page of EDMW and extracted user replies by identifying HTML div containers with a specific class. Similarly, after filtering, around 8,000 unique comments from EDMW about a wide range of discussion topics.

2.3 Data Preprocessing

2.3.1 Common Techniques

The project referenced data preprocessing techniques from other state-of-the-art NLP models. Regular expression (regex) patterns were used to format the forum comments to provide the labelling functions with relevant information. These patterns removed non-essential content such as URLs, email addresses, user mentions and non-alphanumeric characters (Chinese/Korean characters and diacritics). Metadata present in EDMW comments contain the comment timestamp, device or application identity and user reactions, which were also deleted. Comments had their letter casing preserved as most comments had the correct capitalisation for named entities. Emojis were converted to words with corresponding semantic meaning using the Python Emoji library [10] to preserve information.

2.3.2 Handling Singlish

Most of the comments from r/Singapore and EDMW were not written in Standard English and were grammatically incorrect. Singlish, a colloquial English-based creole with elements of Chinese dialect and Malay, along with Internet slang and abbreviations, can often be found in these comments . Hence, regex patterns from a custom dictionary replaced various site-specific lingo and terms, especially for EDMW. This step is needed to enable the labelling functions pre-trained on Standard English corpora to recognise proper nouns and overall sentence structure. In addition, Singlish sentence-final particles used for exaggeration like “lah”, “hah”, and “ah” were removed as they do not affect the sentence structure for NER and OLI tasks [11].

2.3.3 Handling Tweets

Comments that OLID provided were compiled from Twitter messages or “tweets” made by Americans. Most of these tweets were written in a different context and format than local forum comments; separate preprocessing steps were applied to OLID comments. Word segmentation was carried out on hashtags by capital letters recognition using the Python WordSegmentation library [12], where #KeithEllisonAbuse would be transformed to “keith ellison abuse”. Redacted user mention tokens in the form of “@USER” were preserved, and multiple mentions in the same tweet were condensed into “@USERS” [13].

These tokens often appeared in tweets and provided contextual clues on the target of the tweet, a key feature employed for the OLI task.

2.4 Labelling Data

Comments from r/Singapore, EDMW and OLID were used for the OLI task. However, only comments from r/Singapore were required for the NER task as most comments from the excluded sources did not contain any named entities. Crowdsourced annotators labelled 13,240 comments in the OLID testing dataset. One thousand comments from r/Singapore and another 1,000 from EDMW were chosen and manually labelled with ground truth labels. The process was still highly time-consuming despite running the comments through existing NER and OLI models to generate estimated labels. On average, identifying named entities in a comment and classifying it based on the OLID tasks took about a minute. This process is unsustainable for larger datasets; there is strong motivation to look into efficient techniques like weak supervision, where the creation of labelled data is automated and models of comparable performance can be developed.

Chapter 3

Model Development

3.1 Libraries Used

3.1.1 spaCy

spaCy [14] is a free, open-source library that provides pre-trained pipelines for different NLP tasks. spaCy was chosen over other libraries like NLTK and Snorkel due to its support for custom-named entity labels, state-of-the-art accuracy benchmarks and relative ease of use. spaCy's RoBERTa-based transformer model has F-scores of 89.8 on the OntoNotes 5.0 corpus and 91.6 on the CoNLL 2003 corpora [15]. spaCy's pipeline could efficiently process large amounts of comments to produce "Doc" objects, containers for accessing linguistic annotations and word tokens. The transformer model created dense, context-sensitive representations for the tokens in the comments. This feature is absent in word vector representation models and generally increases the performance of models when evaluated on NER and OLI tasks. This project fine-tuned the spaCy transformer model on labels generated through weakly supervised labelling functions for both tasks. The weak supervision models' performance was compared to similar transformer models fine-tuned on ground truth labels.

3.1.2 skweak

This project used skweak [16], a Python toolkit for applying weak supervision to NLP tasks, to generate labelled training data without manually creating ground truth labels. The toolkit defines and aggregates labels from default labelling functions such as pre-trained machine learning models, gazetteers, crowd-worker annotations, and domain-specific heuristics. These labels are then resolved to a single, probabilistic label for each entity by estimating a hidden Markov model using the Baum-Welch algorithm [17]. The spaCy transformer models were then fine-tuned using these resolved labels.

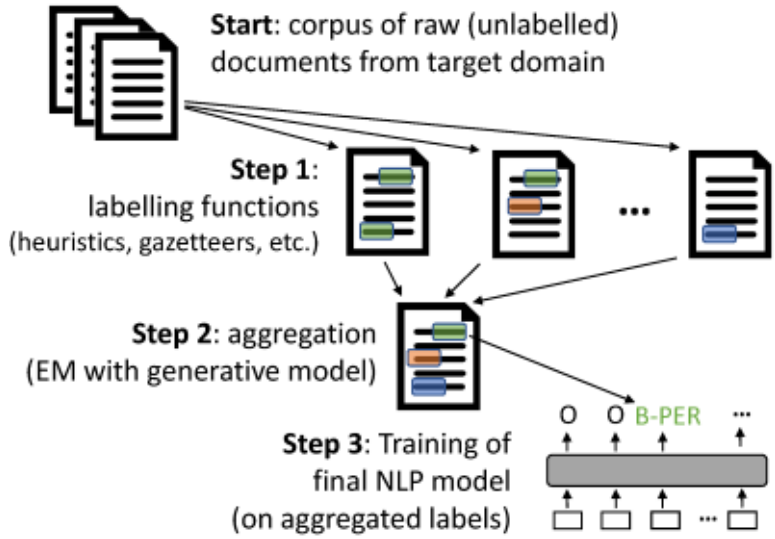

Figure 3.1: Basic overview of skweak

For the NER task, the spaCy transformer model needed to identify named entities in the following CoNLL-2003 categories: "Location", "People", "Organisation", and "Miscellaneous". skweak could standardise the eighteen different named entity labels from spaCy OntoNotes models into these four categories. For example, named entities identified as "Events", "Product", and "Language" by the spaCy model were reclassified as "Miscellaneous". Additionally, "Geopolitical Entities", which included countries and cities, were now tagged as "Locations".

The labelling functions used for this NER task included the base spaCy transformer model's NER component and skweak's knowledge bases from Wikidata and Crunchbase. Custom labelling functions, which used rule-based heuristics and local database lookups, were implemented to identify local entities. MRT and LRT station names, along with neighbourhoods and major roads in Singapore, were compiled to identify "Location" entities that skweak's components cannot recognise. Additionally, skweak's heuristic functions tend to identify capitalised abbreviations, such as "ICU" and "WFH", as "Organisations". Hence, negative labels of "Not-Organisations" were applied to these non-entities, a helpful feature in skweak.

The weights of these labelling functions were determined after analysing each component's F-score. These weights affected skweak's aggregator function in resolving conflicted or ambiguous entities as it uses the highest-scoring label. Labelling functions that identified frequent and unambiguous entities, like "PAP", were given greater weights as the abbreviation could only refer to Singapore's People's Action Party. Conversely, a phrase like "Sia" could refer to the initials for Singapore Airlines (SIA), the Australian pop singer, or simply a sentence-final particle. This issue was challenging to resolve as comments did not use proper capitalisation.

3.1.3 Hugging Face Transformers

The OLI problem was approached as a multi-category classification task. First, the model identifies if the comment is offensive or not (Task A). An offensive comment is defined as an inflammatory statement that provokes a response. Next, the model deduces if the offensive comment it identified is targeted (Task B). Lastly, if the offensive comment is targeted, the model determines if the insult or threat is targeted at an individual, group or other entity (Task C). These sequential annotation schema was suggested by OLID and described in more detail in Table 3.1. The models were benchmarked using OLID comments against the results of a Convolutional Neural Network (CNN) model presented in the OLID report [18].

Comment	**Task A**	**Task B**	**Task C**
He is a friendly person.	Not Offensive	-	-
Worst trip of my fucking life.	Offensive	Untargeted	-
Screw this fat cock sucker.	Offensive	Targeted	Individual
What is wrong with these idiots?	Offensive	Targeted	Group
AWARE is an organisation for losers.	Offensive	Targeted	Other

Table 3.1: Example of OLI annotation schema

skweak was unable to output multi-category classification labels for comments. Hence, this project used the Hugging Face Transformers library [19] to generate the weak supervision labels used for the OLI task. An ensemble of pre-trained OLI models from the library, sentiment analysis models and offensive language lexicons served as labelling functions for Task A. Selective OLI models that could distinguish insults from hate speech were employed for Task B. For Task C, a custom script analysed the presence and type of named entities, proper nouns and subject of the comment. The weak supervised NER model developed in this project helped ascertain the target of the comment. Similar to the NER task, the weights of the components were adjusted based on their individual F-scores. Sentiment analysis components were given lower weights as a negative comment may not necessarily be offensive. Conflicting labels were resolved based on scores to obtain the final labels used for fine-tuning. Even with GPU batch processing, generating weak labels for about 13,240 OLID comments was the slowest weak supervision technique and roughly took an hour.

3.2 Fine-tuning

To fine-tune the spaCy transformer models, comments in each dataset were split into training, validation and testing sets in a 60%-20%-20% manner. Fine-tuning the non-weak supervision model was done using pseudo-rehearsal. Labels from the base model's pipeline output or revision labels were to be included in fine-tuning. As such, 30% of the comments in training and validation sets were randomly chosen to be tagged with revision labels instead of ground truth labels. This approach prevents catastrophic interference, a phenomenon where a pre-trained model forgets its previously learnt knowledge and transfer learning cannot occur [20].

A Python script using the spaCy pipeline processed comments in the training and validation sets and their corresponding labels into binary files. These .spacy files were then used to fine-tune a spaCy transformer model from the command line using default hyper-parameters. Fine-tuning was performed on a laptop equipped with an NVIDIA Quadro RTX 5000 GPU and 64GB of RAM. On average, fine-tuning took about an hour for NER models and thirty minutes for OLI models. The number of comments in the dataset had a negligible impact on the duration.

Chapter 4

Results and Findings

4.1 Evaluation Metrics

This project relied on precision, recall and F1 metrics to assess the accuracy of models on the test set for both tasks. The following equations define the metrics:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{4.1}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{4.2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.3}$$

4.2 NER

4.2.1 Evaluation Criteria

Detected named entity spans must match the right entity category to be counted as a true positive. If the detected span is "Mr John Doe" while the ground truth label span is "John Doe", the detected entity is treated as a false positive. Although there are imbalances in the type of entities in the testing set, as seen in Table 4.1, the micro average F1 score was measured to incentivise the model to maximise its number of correct predictions [21].

Entity	Count	Percentage
Location	144	40.4%
Miscellaneous	111	31.2%
Organisation	76	21.3%
People	25	7.1%
Total	**356**	**100.0%**

Table 4.1: Entity distribution in 200 r/Singapore comments

4.2.2 Results

Three NER models based on the spaCy transformer model were evaluated on 200 r/Singapore comments in the test set; the weak supervision model was compared against the model fine-tuned on ground truth labels and a base model that did not undergo fine-tuning. As the base model was trained on the Ontonotes corpus, its eighteen named entity categories were condensed to fit the four categories from the CoNLL-2003 scheme. The metrics of the three cased models are shown in Table 4.2.

r/Singapore comments

Fine-tuned model	Precision	Recall	Micro F1
Weak Supervision labels	80.1	81.2	80.6
Ground truth labels	85.3	83.7	84.5
Base	84.5	78.9	81.6

Table 4.2: NER performance on r/Singapore comments

4.2.3 Findings

Compared to evaluations done on cleaner datasets and corpora, all three models performed worse due to inherent data noise from misspellings and grammatical errors in online forum comments [13]. The weak supervision model's accuracy was within 5% of the model fine-tuned on ground truth labels. Despite the base transformer model having a higher F1 score than the weak supervision model, the latter had a higher recall for local named entities. As seen in Figure 4.1, the weak supervision model had higher sensitivity towards identifying instances of religious groups, which the other models omitted. Figure 4.2 shows the weak supervision model identifying companies spelt in lowercase. In addition, the model could distinguish the difference in context between sentences with the term "Tiananmen" in Figure 4.3.

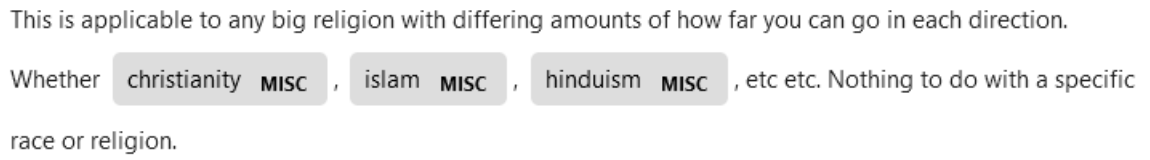

Figure 4.1: Identifying religious groups

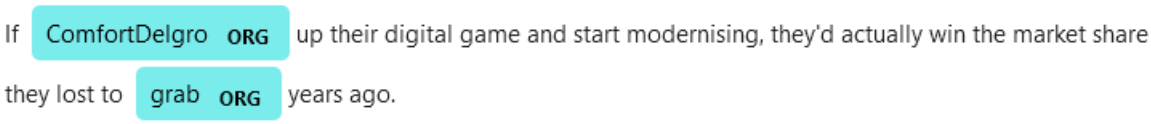

Figure 4.2: Identifying companies spelt in lowercase

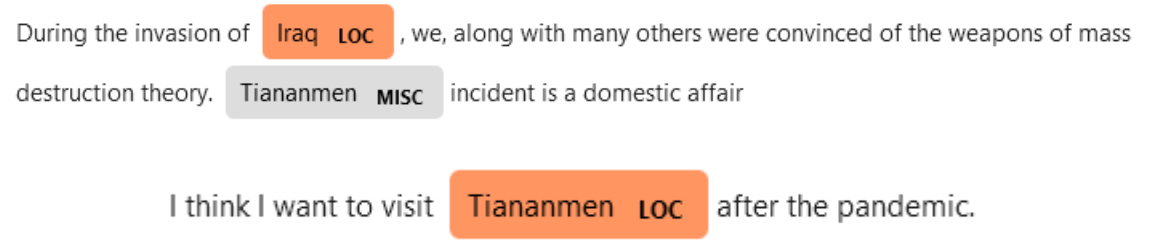

Figure 4.3: Distinguishing entity classes based on context

4.3 OLI

4.3.1 Evaluation Criteria

The fine-tuned models output a confidence score between 0 and 1 for OLI Tasks A and B. In contrast, the model chose the most likely target for Task C. Task A used the entire testing set to evaluate the model's ability to identify offensive comments. However, Task B only used comments with a ground-truth label of offensive, while Task C used a subset of these comments that were targeted. The distribution of sentence types is detailed in the results for the different datasets.

4.3.2 Results

For the three OLI tasks, the models were evaluated on comments from r/Singapore, EDMW and OLID. The same 1,000 r/Singapore comments from the NER task were used. 1,000 comments from EDMW were also included. For OLID, a model used 13,240 comments for training and validation, with a separate dedicated testing set that contains 860 comments. The project also explored the performance of NLP models when fine-tuned using a smaller dataset. Models fine-tuned with only 1,000 out of these 13,240 comments were dubbed "Small". A Python script determined the optimal threshold for each model to obtain its maximum F1 Macro score in Tasks A and B. The performance of each model on different datasets is shown in the tables below. The frequency of the categories based

on the annotation schema is also detailed in brackets for each dataset.

OLID comments

Fine-tuned model	**F1 OFF (240)**	**F1 NOT (620)**	**Macro F1 (860)**
Weak Supervision labels	72.5	90.0	81.2
Weak Supervision labels (Small)	69.6	87.9	78.7
Ground truth labels	71.9	89.1	80.5
Ground truth labels (Small)	65.2	86.1	75.6
CNN	70.0	90.0	80.0

Table 4.3: Task A performance on OLID tweets

Fine-tuned model	**F1 TIN (213)**	**F1 UNT (27)**	**Macro F1 (240)**
Weak Supervision labels	91.4	28.6	60.0
Weak Supervision labels (Small)	91.4	28.6	60.0
Ground truth labels	80.0	30.2	55.1
Ground truth labels (Small)	94.0	24.2	59.1
CNN	92.0	42.0	67.0

Table 4.4: Task B performance on OLID tweets

Fine-tuned model	**F1 IND (100)**	**F1 GRP (78)**	**F1 OTH (35)**	**Macro F1 (213)**
Weak Supervision labels	77.3	61.9	25.9	55.0
Weak Supervision labels (Small)	74.7	56.2	17.3	49.4
Ground truth labels	72.6	46.7	27.7	49.0
Ground truth labels (Small)	71.4	43.4	17.8	44.2
CNN	75.0	67.0	0.0	47.0

Table 4.5: Task C performance on OLID tweets

EDMW comments

Fine-tuned model	F1 OFF (116)	F1 NOT (84)	Macro F1 (200)
Weak Supervision labels	87.3	81.3	84.3
Ground truth labels	87.7	84.4	86.1

Table 4.6: Task A performance on EDMW comments

Fine-tuned model	F1 TIN (91)	F1 UNT (25)	Macro F1 (116)
Weak Supervision labels	86.8	46.5	66.6
Ground truth labels	88.5	60.0	74.3

Table 4.7: Task B performance on EDMW comments

Fine-tuned model	F1 IND (36)	F1 GRP (44)	F1 OTH (11)	Macro F1 (91)
Weak Supervision labels	62.7	39.3	16.7	39.6
Ground truth labels	68.4	79.1	0.0	49.2

Table 4.8: Task C performance on EDMW comments

r/Singapore comments

Fine-tuned model	F1 OFF (62)	F1 NOT (138)	Macro F1 (200)
Weak Supervision labels	65.5	87.2	76.3
Ground truth labels	63.7	86.4	75.1

Table 4.9: Task A performance on r/Singapore comments

Fine-tuned model	F1 TIN (52)	F1 UNT (10)	Macro F1 (62)
Weak Supervision labels	91.1	66.7	78.9
Ground truth labels	86.0	48.0	67.0

Table 4.10: Task B performance on r/Singapore comments

Fine-tuned model	F1 IND (18)	F1 GRP (19)	F1 OTH (15)	Macro F1 (52)
Weak Supervision labels	64.0	46.7	25.0	45.2
Ground truth labels	73.7	48.5	42.4	54.9

Table 4.11: Task C performance on r/Singapore comments

4.3.3 Findings

Similar to the NER task, the weak supervision model performed comparably to the model fine-tuned on ground truth labels on all three datasets. The fine-tuned models performed better than the CNN model on Tasks A and C when evaluated on OLID comments. The "Small" models were fine-tuned and evaluated on a limited set of 1,000 comments, resulting in a decrease in accuracy across the three tasks. This observation suggests that weak supervision on a larger dataset could result in a more accurate model than a smaller manually labelled dataset. Category imbalance was apparent in this dataset, and categories with limited comments had drastic decreases in F1 scores. This issue is likely inherent due to online forums censoring offensive and hateful content. EDMW replaces offensive words like "fuck" with asterisks, and automoderators on r/Singapore actively delete offensive comments that breach the subreddit rules. Hence, the weak supervision models may not be able to identify targeted threats and comments encouraging crime if the training dataset lacks content in these aspects.

Chapter 5

Web Application

5.1 Deployment

This project created a web application interface to demonstrate the weak supervision models' ability to perform NER and OLI on unseen text. The web application uses the Axios client [22] to host a web server which handles HTTP requests to and from the application server. The backend application server was created using the Flask microframework [23] and its cross-origin resource sharing library [24]. Users can select an OLI model based on the dataset used to fine-tune it. The chosen OLI model, along with the NER model fine-tuned on r/Singapore comments, processes the input sentence with fixed thresholds for OLI Tasks A and B. The web application colour codes detected named entities using the displaCy visualiser [25], and the three OLI categories of the sentence are shown. The project implemented client and server-side input validation and a loading overlay to indicate that the model was processing the sentence. A hundred word sentence takes the application about nine seconds to display results, as seen in Figure 5.1.

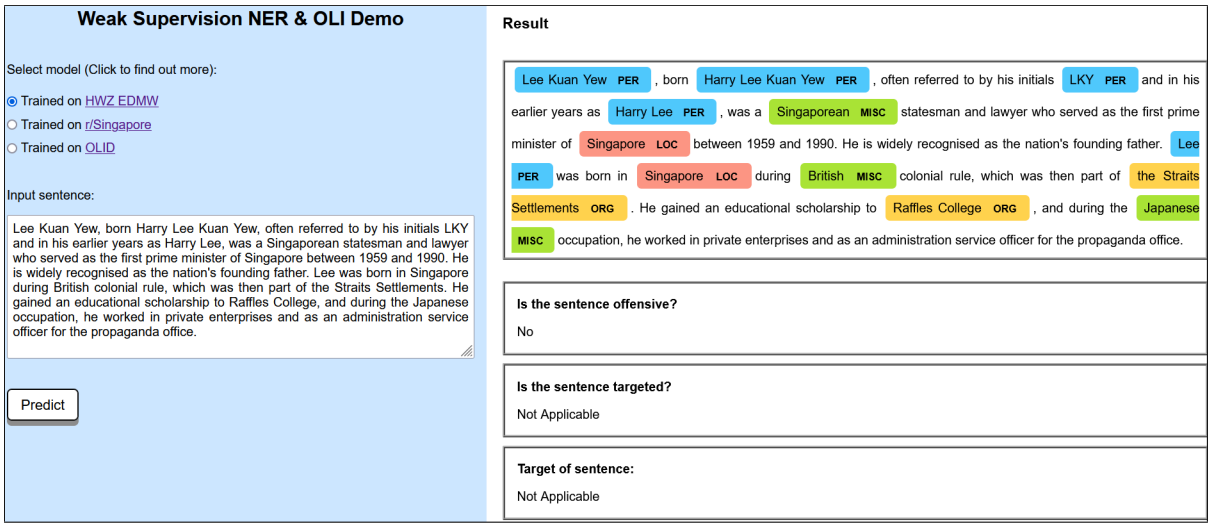

Figure 5.1: Interface of web application

5.2 Findings

The NER and OLI weak supervision models could identify named entities and offensive content in unseen text. In Figure 5.2, the OLI model fine-tuned on EDMW comments correctly identified an unseen EDMW comment as offensive. However, the NER model wrongly identified the named entities in the comment.

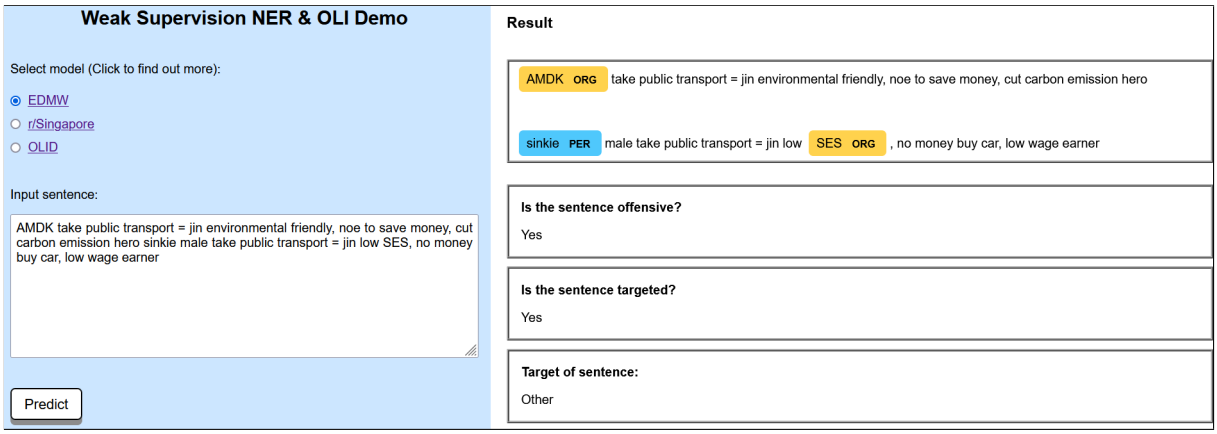

Figure 5.2: Unseen comment from EDMW

The OLI model fine-tuned on OLID comments was able to identify the "Navy Seal Copypasta" as offensive. In addition, the NER model correctly detected most of the named entities in the paragraph, even with the terrorist organisation "Al-Qaeda" being misspelt, as seen in Figure 5.3.

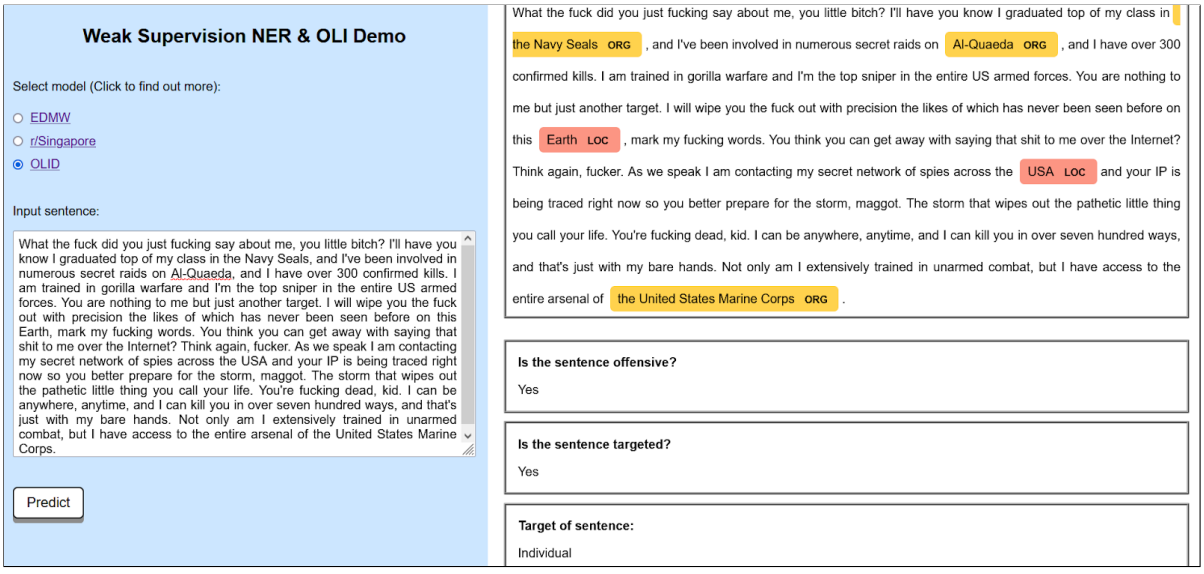

Figure 5.3: Unseen Navy Seal Copypasta

Chapter 6

Conclusion and Recommendations

6.1 Conclusion

This project's weak supervision models performed comparably on NER and OLI tasks to transformer models fine-tuned on ground truth labels. Weak supervision is a feasible alternative to generating labelled training data, especially when manually labelling a large dataset is too time-consuming. Users can adjust labelling functions easily to suit their needs. The user can optimise the model using the recall metric for OLI tasks as false-positive cases of offensive sentences are more acceptable than false-negative cases.

6.2 Future Work

The project can be extended to make the language models multilingual or language agnostic. This feature will be handy for analysing online comments from Singapore-based forums. A noticeable amount of forum comments contain Chinese or Malay words, which the English-based models currently ignore. A further step for the NER task would be to create a Named Entity Linking model, where the named entities detected are assigned a unique identifier such as a Wikipedia page. This extension could be helpful for educational purposes, where the user learns more about specific people or organisations from its corresponding Wikipedia article. For the OLI task, linguistic and psycholinguistic analysis techniques could be included as a weak signal. Targeted offensive sentences tend to be informal in an angrier and authoritative tone and contain fewer analytical words [26]. Weighted labelling functions can be created to identify these sentence features for OLI Task B and its extent of correlation. Lastly, the web application could include an upload function that can handle files in PDF or CSV format. This feature would make it more convenient for the user to analyse multiple sentences without manually transferring the file contents into the input field.

References

[1] Harshith Nadendla. *Why are LSTMs struggling to matchup with Transformers?* URL: https://medium.com/analytics-vidhya/why-are-lstms-struggling-to-matchup-with-transformers-a1cc5b2557e3.

[2] Alexander Wissner-Gross. *Datasets Over Algorithms*. URL: https://www.edge.org/response-detail/26587.

[3] Alex Ratner et al. *Weak Supervision: A New Programming Paradigm for Machine Learning*. URL: https://ai.stanford.edu/blog/weak-supervision/.

[4] Erik F. Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 2003, pp. 142–147. URL: https://aclanthology.org/W03-0419.

[5] *Singapore*. URL: https://www.reddit.com/r/singapore/.

[6] *Eat-Drink-Man-Woman | HardwareZone Forums*. URL: https://forums.hardwarezone.com.sg/forums/eat-drink-man-woman.16/.

[7] Marcos Zampieri et al. *OLID*. URL: https://sites.google.com/site/offensevalsharedtask/olid.

[8] *PRAW: The Python Reddit API Wrapper*. URL: https://praw.readthedocs.io/en/stable/.

[9] *Beautiful Soup*. URL: https://www.crummy.com/software/BeautifulSoup/.

[10] *emoji 1.7.0*. URL: https://pypi.org/project/emoji/.

[11] Siaw Ling Lo et al. "A Multilingual Semi-Supervised Approach in Deriving Singlish Sentic Patterns for Polarity Detection". In: *Know.-Based Syst.* 105.C (Aug. 2016), pp. 236–247. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2016.04.024. URL: https://doi.org/10.1016/j.knosys.2016.04.024.

[12] *wordsegment 1.3.1*. URL: https://pypi.org/project/wordsegment/.

[13] Wenliang Dai et al. "Kungfupanda at SemEval-2020 Task 12: BERT-Based Multi-TaskLearning for Offensive Language Detection". In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 2060–2066. DOI: 10.18653/v1/2020.semeval-1.272. URL: https://aclanthology.org/2020.semeval-1.272.

[14] *spaCy · Industrial-strength Natural Language Processing in Python*. URL: https://spacy.io/.

[15] *Facts & Figures · spaCy Usage Documentation*. URL: https://spacy.io/usage/facts-figures.

[16] *skweak: A software toolkit for weak supervision applied to NLP tasks*. URL: https://github.com/NorskRegnesentral/skweak.

[17] Pierre Lison, Jeremy Barnes, and Aliaksandr Hubin. "skweak: Weak Supervision Made Easy for NLP". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Aug. 2021, pp. 337–346. DOI: 10.18653/v1/2021.acl-demo.40. URL: https://aclanthology.org/2021.acl-demo.40.

[18] Marcos Zampieri et al. *Predicting the Type and Target of Offensive Posts in Social Media*. 2019. DOI: 10.48550/ARXIV.1902.09666. URL: https://arxiv.org/abs/1902.09666.

[19] *Hugging Face – The AI community building the future*. URL: https://huggingface.co/.

[20] Matthew Honnibal. *Pseudo-rehearsal: A simple solution to catastrophic forgetting for NLP*. URL: https://explosion.ai/blog/pseudo-rehearsal-catastrophic-forgetting.

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. "Introduction to Information Retrieval". In: Cambridge University Press, 2008. Chap. 13.

[22] *axios: Promise based HTTP client for the browser and node.js*. URL: https://github.com/axios/axios.

[23] *flask: The Python micro framework for building web applications*. URL: https://github.com/pallets/flask.

[24] *Flask-CORS – Flask-Cors 3.0.10 documentation*. URL: https://flask-cors.readthedocs.io/en/latest/.

[25] *displaCy · spaCy Universe*. URL: https://spacy.io/universe/project/displacy.

[26] Mai ElSherief et al. *Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media*. 2018. DOI: 10.48550/ARXIV.1804.04257. URL: https://arxiv.org/abs/1804.04257.