UNT Linguistics at SemEval-2020 Task 12: Linear SVC with Pre-trained Word Embeddings as Document Vectors and Targeted Linguistic Features

Jared Fromknecht and **Alexis Palmer**
Department of Linguistics, University of North Texas
jaredfromknecht@my.unt.edu
alexis.palmer@unt.edu

Abstract

This paper outlines our approach to Tasks A & B for the English Language track of SemEval-2020 Task 12: OffensEval 2: Multilingual Offensive Language Identification in Social Media. We use a Linear SVM with document vectors computed from pre-trained word embeddings, and we explore the effectiveness of lexical, part of speech, dependency, and named entity (NE) features. We manually annotate a subset of the training data, which we use for error analysis and to tune a threshold for mapping training confidence values to labels. While document vectors are consistently the most informative features for both tasks, testing on the development set suggests that dependency features are an effective addition for Task A, and NE features for Task B.

1 Introduction and System Overview

SemEval 2020 Task 12: Offenseval 2 (Zampieri et al., 2020) is an offensive language identification task revolving around classifying social media comments. For the English track, there are three sub-tasks: Offensive Language Identification (Task A), Offense Type Categorization (Task B), and Offense Target Identification (Task C). We focus on Tasks A & B. In contrast to 2019's task, this year's training data is labeled using distant supervision: various supervised models trained on the manually-annotated OLID-2019 data set (Zampieri et al., 2019a) output labels and confidence values for a much larger data set, resulting in the SOLID dataset (Rosenthal et al., 2020). The semi-supervised labels are converted to an average confidence value between 0 and 1; 1 is aligned to the positive class for the task and 0 to the negative class. Thus an additional challenge in this task is determining an appropriate threshold value for mapping confidence measures to labels. To this end, we create a held-out hand-annotated development set from the training data for each task and use it as a test set for picking a good threshold.[1]

Using Scikit-learn's Linear SVM implementation (Pedregosa et al., 2011), we build a classifier with features based on two sets of pre-trained word embeddings: GloVe's 200-dimension Twitter embeddings (Pennington et al., 2014) and 200-dimensional word2vec Twitter embeddings (Deriu et al., 2017). Additionally, we explore four categories of linguistic features beyond the embeddings: lexical, part of speech, dependency, and named entity features. We use spaCy (Honnibal and Johnson, 2015) for preprocessing and extraction of linguistic features.

Ablation studies of the additional linguistic features establish which linguistic features are most effective in each task. The goals of these studies are: To further target which features are most informative for each of the sub-tasks; to also narrow in on features that are ineffective or may be detrimental to classification; and to identify which types of offensive constructions are still missed by the model and why. In other tasks, such as multi-lingual sentiment analysis, success has been found in Deep Learning architectures that have utilized both word embeddings and feature embeddings as input (Akhtar et al., 2019). For some tasks, the most appropriate types of feature embeddings may be directly apparent. However, in a complex semantic problem such as recognizing offensive and abusive language, the ideal feature support to the word embedding input is not immediately clear. Feature engineering and error analysis can offer

[1]Code and hand-annotated data sets available via GitHub: https://github.com/jmfromkn/Offenseval_Code

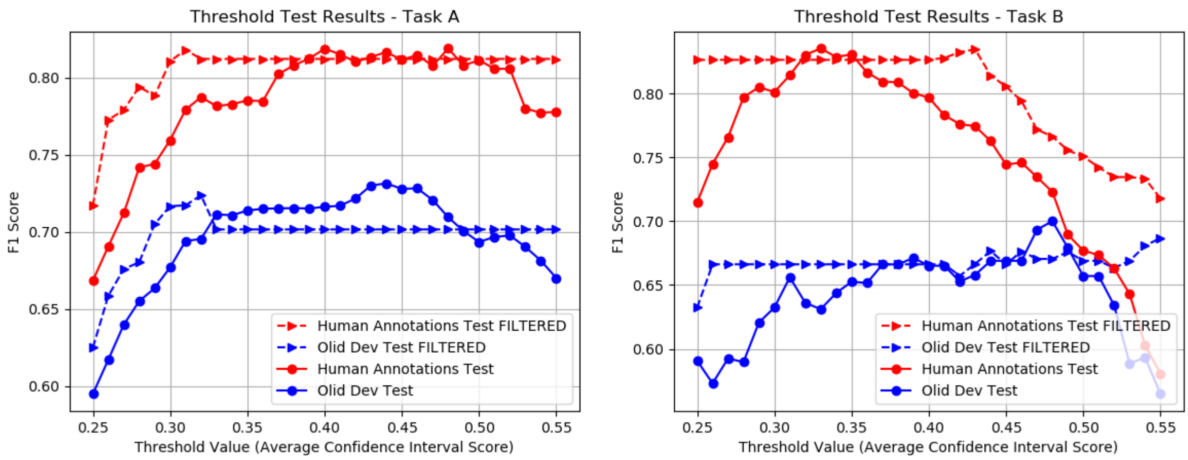
Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2209–2215
Barcelona, Spain (Online), December 12, 2020.

additional information in understanding which linguistic relations best inform feature extraction for future implementations.

2 Data

All experiments are performed using the SOLID (Rosenthal et al., 2020) and OLID (Zampieri et al., 2019a) data sets. The OffensEval annotation scheme is a pipeline: tweets are labeled as OFF or NOT in Task A, and only offensive tweets go on to Task B, where they are labeled as targeted or untargeted. This means good performance on earlier subtasks is essential for good overall performance; thus we focus on the first two subtasks. We randomly sample 1 million tweets from the total training data for Task A (over 9 million tweets), and we use the entire training set for Task B (188,727 tweets). As mentioned above, we create small hand-annotated validation sets for each task: 375 tweets for Task A and 350 for Task B.

Discretization and Threshold Testing. To identify an appropriate threshold for discretizing the average confidence values in the training set (i.e. to convert them to labels of OFF or NOT), we perform iterative testing of different threshold values, using an increment of 0.01 and measuring F1 on the hand-annotated validation sets. One concern with this approach is that all annotations in the hand-annotated data sets come from our first author, with no discussion or adjudication. Thus we perform a second round of iterative testing using last year's OLID test data as validation data. We decide final threshold values by compromising between the best thresholds on our human-annotated data and on the OLID-2019 test data. We try two different strategies for picking the best threshold, described below. Results appear in Figure 1; lines with circles show the simple average strategy, and those with triangles show the filtering strategy.



(a) Task A

(b) Task B

Figure 1: F1 Scores for different threshold values for both subtasks, comparing two different validation data sets and two different selection strategies.

Average Confidence Value Testing. In this strategy, we compare performance on the validation data sets for different thresholds on the average confidence value provided with the training data. The best values testing on our small manually-annotated data set are 0.40 and 0.33 for Tasks A and B, respectively. Testing on OLID-2019, the best threshold values for the two tasks are 0.44 and 0.47, respectively. Compromise values, used for further classification, are 0.42 and 0.38.

Filtering. The second strategy is designed to remove many of the less-confident or borderline cases of the data, leaving only clearer instances of positive and negative classes. To do this, we set two conditions: average confidence values must be either above a certain value or below another value (avoiding instances with scores around the midpoint of the data), and standard deviation for the scores must not exceed a specified value. As such, the following filters were set for Task A&B's training data:

Task A Filter: ((Average ≤ 0.32 | Average ≥ 0.68) & Standard Deviation < 0.18)
Task B Filter: ((Average ≤ 0.25 | Average ≥ 0.4) & Standard Deviation < 0.27)

The filtering strategy has a noticeable effect on performance trends, effectively flattening the peaks and troughs seen with the simpler method. While filtering allows for a more generous range when determining the best threshold, with a larger margin for error, there is a fundamental concern of whether or not this method generalizes well to new test data, having removed all of the difficult and/or vague cases. We use the simple method for our submitted system.

3 Model and Features

Although the most successful systems from Offenseval 2019 were primarily neural systems (Zampieri et al., 2019b), we choose to use a non-neural SVM classifier because of the ease of feature inspection, feature specification, and error analysis. Specifically, we use scikit-learn's linear SVM implementation (Pedregosa et al., 2011) with default parameter values.

Previous SVM-based approaches to offensive language detection investigate the effectiveness of sentence embeddings (Indurthi et al., 2019, for example) and lexical features such as sentiment analysis and offensive/profane word lexicons (Plaza-del Arco et al., 2019, among others). We hypothesize that part of speech, dependency, and named entity features capture additional context useful for classifying tweets with more elusive forms of offensive language. These context-level features could also be useful in Task B, to capture differences in the constructions seen in targeted vs. untargeted tweets. Our base model uses the dimensions of a document embedding (described below) as features; our additional linguistic features are listed in Table 1 and described below.

Feat. Group	List
Lexical	Avg. # Punct, Avg. Token Length, # Tokens, # Non-alpha Subs, @User Front (Binary), @User Back (Binary)
Part of Speech	TFIDF of POS Counts
Dependency	TFIDF of DependencyTag_RootPOS
NE	NE TFIDF, Has Person, Has Organization, Has Nationality, Has Place

Table 1: Linguistic Feature List

Document embeddings. Following Mitchell and Lapata (2010), we build document embeddings (treating each tweet as a document) by averaging the vectors of the individual words in the tweet. For coverage, we concatenate document vectors based on GloVe's 200-dimension Twitter embeddings (Pennington et al., 2014) with document vectors based on twitter-trained English word2vec embeddings (Deriu et al., 2017).

Lexical features. This category of features includes a combination of simple token features, including counts and length, with several twitter specific features. In particular, features related to user names and non-alphabetic character substitutions aim to account for alternate word constructions that may be reflected in the pre-trained embeddings. Davidson et al. (2017) use a similar non-alpha substitution feature when looking at Yahoo! Finance comments due to observations that some users self-censor their profanity.

Part of speech and Dependency features. Along with tokenization, part of speech tagging and dependency parsing was also done by spaCy's NLP pipeline for each tweet. For part of speech features, we build a TFIDF matrix of tags. For dependency features, which describe relation to a root, we concatenate dependency tags with their corresponding root's POS tag to form "Dependency_HeadPOSTag" constructions and, similarly, build a TFIDF matrix. In both cases, we use sub-linear term frequency.

Named entity features. We use spaCy's Named Entity Recognition (NER) parser to produce named entity tags for tweets. The intuition is that NE information should be useful for Task B in particular, as the presence or lack of certain NE constructions could signal a targeted or untargeted tweet. We implement this as four binary features indicating the presence of Person, Place, Nationality, and Organization tags.

Feature Set - add	**Task A F1**	**Task B F1**	**Feature Set - subtract**	**Task A F1**	**Task B F1**
Embeddings Only	0.796	0.821	All Features	**0.819**	**0.835**
Embeddings + NER	0.798	**0.832**	All minus NER	0.816	0.824
Embeddings + POS	0.812	0.8206	All minus POS	0.816	0.827
Embeddings + LEX	0.803	0.812	All minus LEX	**0.819**	**0.835**
Embeddings + DEP	**0.819**	0.809	All minus DEP	0.809	0.823

Table 2: Ablation studies on development sets. Additive (left): embedding features as base model. Subtractive (right): full feature set as base model. Thresholds: Task A = 0.40 and Task B = 0.33

Additionally, to test whether additional NE types could be indicators for Task B, we construct a TFIDF matrix of named entity tags in a given tweet.

4 Analysis

We perform two types of analysis to better understand the role of various linguistic feature sets for offensive language classification: ablation studies (Table 2) and qualitative error analysis (Tables 3 and 4). We do ablation studies in two directions: adding feature sets to our base model, and subtracting them from the full model (base model plus all linguistic feature sets). All results in this section are from testing on our hand-annotated validation set, with best-performing threshold values. For each task, one additional feature set triggers the largest gain in F1, and additional feature set combinations result in negligible gains; we do additional error analysis for these two feature sets. The lexical feature set seems to be the least informative for both tasks, as its addition or subtraction causes minimal changes.

Text	Gold	Base	+Dep
1. "I forgot that for my city school is back in session, and I nearly shat myself when I saw a school bus"	OFF	NOT	OFF
2. Ain't nothing wrong with that 😂Some of you should just accept you was a quick f**k. & that's okay 😂	OFF	NOT	OFF
3. @USER That must have been extremely painful!	NOT	OFF	NOT
4. @USER We need some or her passion in politics. She would wipe the floor with Johnson.	NOT	OFF	NOT
5. Hey I don't like the term bbc or to use it. I just like that I have sex with people. No label necessary.	OFF	NOT	OFF
6. "I've got this love-hate relationship with fluid dynamics 😪 it's so interesting, but so f**king hard 😭 😭"	OFF	NOT	OFF

Table 3: Examples where the addition of dependency features corrects Baseline model errors in Task A.

Task A. Dependency features are the most effective of the linguistic feature sets for Task A, triggering the largest performance gain in additive ablation and the largest drop in subtractive ablation. Part of speech features are the second most informative. Intuitively, context-based features like dependencies and POS tags may capture relations with unknown words or lexical variants missed by the embeddings. For Task A, POS and dependency features seem to overlap, though the results suggest that dependency relations to the POS tag of the head have more informative weight than plain POS tags. The difference between the effect of the two feature sets is minimal in additive ablation but stronger in subtractive ablation. Table 3 shows a sampling of misclassifications made by the embeddings-only (Base) model that are corrected through addition of dependency features. Several of these examples include variations of profanity that may not have listings in the pre-trained embeddings such as "shat" as a replacement for "sh*t" in example 1.

Example 4 illustrates offensive or profanity-based acronyms such as "bbc". In these cases, context-based relations bridge the gap that unknown words or variation create in the embedding baseline. In cases such as examples 2 and 6, while the offensive keywords have a strong association with the OFF label, the presence of emojis may interfere with the mean vector of the embeddings, as these are unlikely to have a concrete entry in the pre-trained embeddings.

Dependency features also have some overlap with other feature categories. One of the twitter-specific lexical features signals the @USER tag in tweet-initial position. @USER at the start of a sentence is parsed as a root dependency tag with no relations. Because of this, the presence of a "Root_POStag" relationship in a sentence commonly marks for start @USER position. Relationships such as these, where the dependency features indirectly measure and have overlap with other feature areas in their context and scope, may explain the size of the effect for dependency features.

Text	Gold	Base	+NE
7. @USER NORMIE! YOU'RE A F**KING NORMIE! GO KILL YOURSELF TO GO TO HEAVEN WITH ALL YOUR FAVORITE PEOPLE THAT DIED!	TIN	UNT	TIN
8. ppl who say animes as the plural form of anime are f**king terrorists	TIN	UNT	TIN
9. leaven big a** branches in my garden	UNT	TIN	UNT
10. @USER Meanwhile Indians are like - what the f**k is Bosnia?	**UNT***	**UNT**	**TIN**
11. Am I the only one that thinks Tiktok cosplayers are weird as f**k	TIN	UNT	TIN

Table 4: Examples where addition of named entity features differs from Baseline in Task B.

Task B. As expected, named entity features offer the most information value on top of the base model for Task B. While specific NE tags could be perceived as being most useful for Task C in distinguishing between Individual, Group, or Other categories, the presence of these aspects can help distinguish between a targeted tweet and an untargeted tweet. We find that the four binary features are more useful than the TFIDF matrix for frequencies of NE tags. The binary features offer sufficient coverage to explain most misclassification errors corrected through the addition of NE features. Some examples appear in Table 4.

A challenging aspect of building the development set for this task is the absence of conversational context. Example 10 could be considered ambiguous as to whether the tweet is targeted, depending on the speaker. If the speaker is Indian (i.e. a member of the potentially targeted group), whether the tweet is targeting that group remains unclear. However, if the tweet is uttered by a person outside that group, it would be more clearly considered a targeted tweet. In the case of this tweet, it gets labeled with Has Place and Has Nationality, suggesting a targeted (TIN) label, but the correct label could change based on conversational context.

Additional features offer very little value, with only the Full Feature model and the Minus Lexical Feature model surpassing the Embeddings-plus-NE model. Though dependency features show greater effectiveness for Task A, they under-perform in Task B, offering less F1-gain when added to the baseline.

5 Results and Future Work

At submission time, our model achieved F1 scores of 0.882 for Task A (Rank 71) and 0.617 for Task B (Rank 16). The model with these scores used thresholds of 0.38 and 0.33 for Tasks A and B. Additional threshold testing on the development set after the initial submission period suggests that further fine-tuning of the threshold may increase F1. Use of the filtering strategy and additional filter parameter testing could also improve performance; we leave investigation of these directions for future work.

To further explore this task, we would like to consider other embedding approaches, considering contextual embeddings such as ELMo (Peters et al., 2018) or large-scale fastText (Bojanowski et al., 2016) embeddings for improving coverage on unknown words and lexical variations common to Twitter.

Another interesting direction is the use of methods to transform emojis into word representation (Singh et al., 2019); we suspect that emojis may play a strong role in conveying offensive language. Since linguistic features seem to be useful for this task, we plan to look into twitter-specific toolkits for extraction of linguistic features; two possibilities are NLTK's tweet tokenizer (Bird et al., 2009) and TweetNLP's part-of-speech taggers and dependency parsers (Owoputi et al., 2013). Finally, larger development sets with more annotators should lead to better threshold testing, especially for Task B.

6 Conclusions

While document-level embeddings are a useful foundation for both Tasks A and B, the usefulness of additional features is highly dependent on the task. Performance on task A, which saw errors where embedding coverage was bypassed by lexical variations, misspellings, or acronyms, is bolstered by context-based features in the form of dependency information. This dependency information also indirectly covered other lexical features and proved to be the most informative of the additional features. Task B is, as hypothesized, most heavily affected by the addition of named entity features.

Acknowledgments

We would like to thank Taraka Rama for coding consultation, helpful conversations, and technical support throughout this project. Thanks also to Eduardo Blanco for his thoughts and input. We would also like to thank the members of the Spring 2020 UNT Linguistics Professional Development course for their continued feedback and support throughout this process. Finally, computational resources were provided by the University of North Texas High-Performance Computing Services, a division of the Research IT Services, University Information Technology, with additional support from UNT Office of Research and Economic Development.

References

Md Akhtar, Abhishek Kumar, Asif Ekbal, Chris Biemann, and Pushpak Bhattacharyya. 2019. Language-agnostic model for aspect-based sentiment analysis. pages 154–164, 01.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging Large Amounts of Weakly Supervised Data for Multi-Language Sentiment Classification. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1045–1052, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September. Association for Computational Linguistics.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Manish Gupta, and Vasudeva Varma. 2019. Fermi at SemEval-2019 task 6: Identifying and categorizing offensive language in social media using sentence embeddings. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 611–616, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380–390.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Flor Miriam Plaza-del Arco, M. Dolores Molina-González, Maite Martin, and L. Alfonso Ureña-López. 2019. SINAI at SemEval-2019 task 6: Incorporating lexicon knowledge into SVM learning to identify and categorize offensive language in social media. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 735–738, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

Abhishek Singh, Eduardo Blanco, and Wei Jin. 2019. Incorporating emoji descriptions improves tweet classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2096–2101.

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. In *Proceedings of NAACL*.

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). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.