An approach framework of transfer learning, adversarial training and hierarchical multi-task learning - a case study of disinformation detection with offensive text

Shravan Chandra and Bhaskarjyoti Das
PES University, Bengaluru, India
E-mail: shrvanchandr@gmail.com, Bhaskarjyoti01@gmail.com

Abstract. With society going online and disinformation getting accepted as a phenomena that we have to live with, there is a growing need to automatically detect offensive text on modern social media platforms. But the lack of enough balanced labeled data, constantly evolving socio-linguistic patterns and ever-changing definition of offensive text make it a challenging task. This is a common pattern witnessed in all disinformation detection tasks such as detection of propaganda, rumour, fake news, hate etc. The work described in this paper improves upon the existing body of techniques by bringing in an approach framework that can surpass the existing benchmarks. Firstly, it addresses the imbalanced and insufficient nature of available labeled dataset. Secondly, learning using relates tasks through multi-task learning has been proved to be an effective approach in this domain but it has the unrealistic requirement of labeled data for all related tasks. The framework presented here suitably uses transfer learning in lieu of multi-task learning to address this issue. Thirdly, it builds a model explicitly addressing the hierarchical nature in the taxonomy of disinformation being detected as that delivers a stronger error feedback to the learning tasks. Finally, the model is made more robust by adversarial training. The work presented in this paper uses offensive text detection as a case study and shows convincing results for the chosen approach. The framework adopted can be easily replicated in other similar learning tasks facing a similar set of challenges.

1. Introduction
With the social media platforms providing anonymity in an online society, disinformation has become commonplace. Rumour, fake news, propaganda, offensive text, abusive text, hate message etc. are part and parcel of disinformation phenomena happening every moment on social media and they are accepted as realities that we need to live with. Hence, in the past few years, disinformation research has gained increasing prominence as a way to counter and control the menace of disinformation. So far, disinformation detection has been primarily a two dimensional approach [1] of characterization and detection. However, the main challenges in successful characterization are the lack of sufficient labeled and balanced dataset in an ever-changing scenario, hierarchical nature in the taxonomy of all kinds of disinformation and changing goal post for model robustness in the face of adversarial attacks. This set of challenges is a common anti-pattern that plagues current disinformation detection approaches. Existing approaches for offensive text identification also face the same set of challenges. The work described in this...
paper proposes an approach to counter this anti-pattern and adopts offensive text detection as a case-study.

2. Existing work

2.1. Transfer vs. multi-task learning in disinformation detection

Transfer learning\[2, 3\] is a type of sequential multi-task learning (MTL) where the learning from a source domain can be transferred to a destination domain. However, the case of negative transfer\[4\] needs to be guarded against. Transfer learning is closely related to other machine learning techniques such as multi-view learning and MTL. In case of multi-view learning, each view provides a different feature set and the learning objective is to take advantage of these complimentary views. In the case of disinformation detection using text, this may not be readily applicable whereas MTL can be considered a close cousin. Multi task learning is the technique of learning together related tasks. Unless the tasks are related, negative transfer\[5\] between tasks can happen with the individual task objectives interfering with each other. MTL also can be fairly beneficial as it effectively prevents over-fitting and acts as a regularization strategy. In case both options are available, transfer learning and MTL can be compared as below:

(i) MTL is focused about transfer between related tasks whereas transfer learning is focused on transfer between related domains.

(ii) MTL can be visualised as simultaneous transfer learning and transfer learning can be visualised as sequential MTL.

(iii) MTL gives equal importance to all the related tasks being compared whereas the main focus of transfer learning is the task in the destination domain.

(iv) Both use very similar techniques i.e. parameter sharing. In MTL, the shared parameters are alternatively updated and that provides better scope to retain task specific learning compared to sequential update of shared parameters in transfer learning where destination domain parameters may completely over-rule those of source domain.

In case of homogeneous transfer learning, the source and the destination have the same feature space but the difference may exist in marginal distribution, labels or even label distribution. In case of heterogeneous transfer learning\[6\], the feature spaces of source and destination may differ and as a result, some domain adaptation strategy may need to be adapted. In the domain of disinformation detection, it will be typically a case of homogeneous transfer learning. In Natural Language Processing (NLP), the pre-dominant usage of transfer learning is in the use of pre-trained embedding for words and sentences such as Glove\[7\], fastText model as provided by Facebook\[8\], BERT\[9\], Roberta\[10\], Xlnet\[11\], ULMFiT\[12\], Universal sentence encoder\[13\] etc.

MTL has been commonly used in disinformation detection primarily to get around the problem of lack of labeled data. In the case where related tasks with labeled data are available, they are learnt together such that knowledge transfer happens between simultaneously learning tasks. Researchers have tried this in various learning tasks in NLP i.e. emotion detection\[14, 15\], stance detection\[16, 17\] fake news detection\[18\], rumour detection\[19, 20\], hate detection\[21\], detection of identity bias\[22\] in toxic comments etc. In offensive text detection, pure MTL based work is rare primarily because labels of all related tasks are not available (labels are only provided for offensive nature of text in the available text dataset). In SemEval 2020 task-12 (offensive language detection), W. Dai et al.\[23\] used MTL between the three sub-tasks A, B and C where each of the sub-tasks uses a BERT based implementation.

Usage of transfer learning in the domain of disinformation detection can be roughly categorized as below:

(i) Using transfer learning with pre-trained embedding: Marzieh Mozafari et al.\[24\] used BERT embedding as the main pre-trained embedding method for hate speech detection
with several variations in the way fine-tuning of pre-trained embedding is done. Valeriya Slovikovskaya\[25\] significantly improved on the benchmark performance of Fake News Challenge stage 1 (FNC-1) stance detection task by using various pre-trained embedding. AG d’Sa et al.\[26\] used a combination of BERT and fastText embeddings as input features to CNN and BiLSTM classifiers for toxic message detection. In SEMEVAL 2019 task 6 for offensive language detection, Ping Liu et al.\[27\] essentially employed BERT pre-trained embedding to win the first place in sub-task A, fourth place in sub-task B and eighteenth place in sub-task C. In this work, the problem of offensive text detection is adopted as a case-study and the results achieved by Liu et al.\[27\]are taken as the baseline performance.

(ii) **General purpose embedding for detection system across sub-categories:** Nothing fits this use case better than the overlapping domains of hate, toxicity, profanity etc. There are many small dataset and there is no uniform definitions of sub-categories across them. Marian-Andrei Rizoiu et al.\[28\] used several existing small dataset to transfer knowledge of hate and come out with a general purpose “hate embedding”.

(iii) **Using transfer learning in lieu of MTL with a related task:** To the best of our knowledge, there are no published work where transfer learning has been used in lieu of MTL with a related task in this domain with significant efficiency.

### 2.2. Hierarchy in disinformation taxonomies

As disinformation research is evolving, the taxonomies for different types of disinformation are constantly evolving as well. This has led to hierarchy in related taxonomies resulting in hierarchy in related learning tasks and dataset. One can find explicit or implicit hierarchy in many types of dataset and tasks in disinformation research:

(i) **Sentiment:** Affective information can be conveyed and perceived at different levels. The predominant examples are dataset that are labeled for both sentiment and discrete emotions\[29, 30\] that can be thought of as a second level in hierarchy.

(ii) **Propaganda:** Semeval-2020 task 11 for propaganda detection is a hierarchical task\[31\] with a similarly structured propaganda detection dataset. At the first level, text span containing the propaganda has to be detected and second subtask is for detecting the specific propaganda technique used.

(iii) **Persuasion technique in multi-modal content:** Persuasion is a key component in all disinformation campaigns. Recently Semeval 2021 task 6\[32\] has been structured around detection of persuasion techniques. The tasks are hierarchical in nature i.e. detecting persuasion text span in the meme, the corresponding technique used for persuasion and persuasion technique detected in the combined image and text.

(iv) **Rumour:** Rumour detection tasks along with dataset such as Pheme\[33\] typically have a two level hierarchy i.e. rumour detection and veracity classification. The corresponding RumourEval task i.e. Semeval 2017 task 8 is essentially a hierarchical task.

### 2.3. Hierarchical multitask learning

Hierarchical classification as a learning task was originally witnessed in information retrieval where very often document classes have sub-classes as well. Alon Zweig et al.\[34\] defined the general framework of hierarchical MTL (HMTL) based on the concept of cascade using the notion of task relatedness. In this framework, at each level, some aspect of the learning problem gets shared. This approach may prove more effective than a simple multi-class or multi-label learning task that effectively flattens the class hierarchy. In this approach, the classification error gets limited to a sub-class and this makes the back-propagated information more effective.

There have been several applications of hierarchical multi task learning in domains such as document classification\[35\], driving\[36\], pedestrian attribute recognition\[37\], facial
attractiveness determination[38] along with face and gender, evaluation of argumentative student essays[39] etc. In all these cases, the related tasks were organized in a hierarchy that was implicit in the domain. Very recently Yang et al.[40] has used a similar approach in a transformer-based implementation for the prediction of the volatility of the financial stock.

2.4. Adversarial Training
The labeled dataset in a problem like offensive language detection is always limited and is a challenge for any supervised learning strategy and is vulnerable to adversarial attack. Miyato et al.[41] proposed a way of regularizing supervised learning by introducing small perturbations in the embedding space. Recently Morris et al.[42] has come up with a framework of text attack implementation that uses various data augmentation strategies to make more robust generalization possible. Deep learning based techniques like Generative Adversarial Network (GAN) and models such as GPT-2[43] can also be used for such purpose.

2.5. Offensive language definition
Paula Fortuna[44] et al. have pointed out the many overlapping terminologies in the areas related to offensive text i.e. offensive language, abusive language, hate speech, cyber-bullying, flaming, toxic comment, profanity, extremism, radicalization etc. The social media platforms also formed their own definition of each terminology while formulating their policies and these definitions differed in the fine prints. Hence, the research in this field is fragmented. Waseem et al.[45] argued that the differences between many of these sub-categories can be figured out if two aspects are considered i.e. whether the content is explicit or implicit and whether it is directed to an individual or group. This definition helped mitigate much of the confusion around the definition of hate speech, cyber-bullying, and offensive language. The Offensive Language Identification Dataset (OLID) [46] in SEMEVAL 2019 Task 6 was formed based on the same criteria.

2.6. Offensive language detection
An important study published in Science [47] found out that disinformation and true information differ in many aspects. Misinformation uses a higher proportion of negative emotion words, exclusivity words (without, but, except etc) etc. The existing work in offensive language identification has been primarily driven by these investigations i.e. the machine learning based models attempt to predict external dimensions or classes based on intrinsic properties or characterizations in the dataset. Offensive language detection and related tasks like hate speech detection in text have been tasks in the Natural Language Processing (NLP) community. The existing efforts in this space have specific themes:

(i) **Purely lexical approach:** This approach is essentially based on the terms or words used. However, even though this approach might be somewhat effective[48] on OLID corpus, it need not be equally effective on all dataset due to the obfuscation strategy used by the users.

(ii) **Classical machine learning based approaches:** Combination of linguistic, knowledge-based and multi-modal features are used by researches[49, 50, 51] along with classical machine learning approaches.

(iii) **Neural network based approach:** Most of these researches [52, 53, 54] used pre-trained embedding along with various neural models. These approaches typically use the pre-trained embeddings such as GloVe[7], Word2Vec[55] etc. in the initial phase. Currently, the transformer-based pre-trained embedding is found to perform better. In fact, out of the top ten teams in SemEval 2019-Task 6, seven teams used BERT[9]. Hence, it was decided to use a transformer-based pre-trained embedding as an option in this work as well.
Table 1: Dataset Description

| Dataset Info            | Task A | Task B | Task C |
|-------------------------|--------|--------|--------|
|                         | Offensive | Not Offensive | Targeted | Un-targeted | Individual | Group | Other |
| Original OLID           | 4880   | 10080  | 4302   | 578        | 2607      | 1230  | 465   |
| Original SOLID          | 1449951 | 7643086 | 150400 | 40496      | 120910    | 22366 | 7123  |
| SOLID Balanced          | 24000  | 24000  | 12000  | 12000      | 4000      | 4000  | 4000  |
| Adversarial Balanced    | 30000  | 30000  | 15000  | 15000      | 5000      | 5000  | 5000  |

3. Dataset
As OLID has been the benchmark dataset for offensive text identification 2019 onward, the same dataset has been chosen for the case study described in this work. Additionally this dataset offered a three level hierarchy instead of two-level as in few other disinformation dataset. The hierarchy in this dataset contains “Offensive”(1) and “Not Offensive”(0) tweets. If a tweet is offensive, it is further categorized as to whether it is “Targeted”(1) or “Not Targeted”(0). Further categorization is done if the tweet is identified as targeted, as to if it is targeted to an Individual(1), or if it is to a Group(2), or any Other(3).

The peculiarities of OLID dataset are as listed below:

(i) The OLID dataset has a hierarchical annotation scheme.
(ii) In a recent work, Caselli et al.[56] pointed out that the annotation scheme in OLID does not annotate the degree of explicitness which is instrumental in differentiating abusive language from offensive language and they have re-annotated the OLID dataset with explicitness information to use it in AbuseEval v1.0 task. Hence a pure lexical approach may be successful with this dataset. Pederson et al.[48] in a recent work, adopted a pure lexical approach and delivered surprisingly high prediction accuracy. A simple TF-IDF based identification of the top 50 keywords points out this fact. This is not surprising as most offensive texts have a familiar set of offensive words.
(iii) The dataset is very imbalanced at every level of the hierarchy. The number of tweets available to represent each class is mentioned in Table 1. To address the imbalance, Rosenthal et al.[57] came up with a larger semi-supervised dataset for offensive language identification(SOLID) that is formed using the base classifier trained with the OLID dataset. Hence the SOLID dataset is a much larger dataset built using weak supervision.

4. Methodology
The methodology adopted here as shown in Figure 1 forms the suggested framework that can be adopted by similar hierarchical classification tasks in disinformation research. The reasoning behind this framework or workflow is listed below:

(i) **Pre-trained embedding alone will not be sufficient for class leading performance:** Gupta et al.[58] has compared the relative performance of different embedding in hate speech detection tasks in text and pointed out a that domain-specific embedding performed slightly worse than domain agnostic embedding. They also pointed out that domain agnostic embedding performs slightly better in dealing with class imbalance in the dataset. Hence
in our work, the OLID dataset being class imbalanced, it was clear that using pre-trained word embedding alone would not provide class-leading performance.

(ii) **Address imbalance of the dataset:** Any text classification task will suffer if the training
is done on an imbalanced dataset. On the other hand, almost all hierarchical classification dataset will be imbalanced. The balance can be achieved by making multi-task learners utilize weak supervision i.e. additional pseudo-labeled dataset generated by a classifier trained on the available dataset. In this particular case, such a dataset pseudo-labeled by a learner trained on OLID dataset was already available and it is utilised to address the imbalance. This additional dataset used here is Semi-Supervised Offensive Language Identification Dataset (SOLID)[57]. The data available for each category is detailed in Table 1, but due to the deletion of old tweets, the numbers quickly fell, and the final numbers are mentioned in Table 1. In the absence of such an existing dataset, a similar weak supervision strategy should be adopted.

(iii) **Adversarial training:** This can be a strategy to further generate additional data (while not disturbing the dataset balance) to bolster the learner against adversarial attacks. However, there is a caveat that needs to be taken care of. If the original dataset is amenable to pure lexical approach (it has typical words for each class), then adversarial training as implemented in this work, may not achieve noticeable improvement in accuracy. This step has to be taken after analysis of frequent words in specific classes. In the case of offensive text, this is the scenario i.e. typical words get used in offensive texts. In this work, adversarial attack is implemented as an experiment and no noticeable improvement in accuracy is seen due to this reason.

(iv) **Choice of pre-trained embedding:** As Malmasi et al. [59] has shown, detecting offensive text such as profanity and hateful text is considerably more challenging than many other NLP classification tasks. The main reason is: hate speech is often very subtle and cannot be easily identified by surface features such as n-grams or count based features such as TF-IDF. In such scenario, learning task requires features that use a deep understanding of text. Primarily because of this consideration, BERT pre-trained embedding is chosen. Note that choice of embedding has to be a conscious decision after considering subtlety in the text data.

(v) **Transfer learning in lieu of multi-task learning for related tasks:** It is hard to obtain dataset labeled for all related tasks in such a setup. In this particular case, offensive language, explicit or implicit, is often accompanied by sentiment and emotion. Since the dataset is not labeled for sentiment, a transfer learning strategy is used to pre-train the model on a sentiment learning task and re-train the same again on an offensive language identification task.

(vi) **Hierarchical multi-task learning with hard parameter sharing:** In this case, sub-task A, B, and C are clearly related and at the same time are hierarchical in nature. So hierarchical MTL is adopted. Hard parameter sharing relies on shared layers between the tasks and that is the methodology that is used in this work while multi-tasking. In this, relative importance of the related tasks is an hyper parameter.

**Algorithm 1** Pseudocode for disinformation detection with an imbalanced hierarchical dataset

1. Address the imbalance in the dataset with adversarially generated data
2. Further augment the dataset with an additional dataset with weak supervision, if available
3. Clean the combined dataset
4. Choose a domain specific or domain agnostic embedding based on dataset vocabulary
5. Split into train and test
6. Train a model on a related learning task for transfer learning
7. Use the above pre-trained model for hierarchical multi-task learning
5. Dataset pre-processing and model
The following pre-processing steps are performed

(i) The tweets were cleaned by removing the stop-words, lower casing all the words, and converting the emoticons into their respective meaning. In this particular case, this is done for all parts of data i.e. the original OLID, samples borrowed from the SOLID and samples generated using augmentation or adversarial techniques.

(ii) Adversarial training is implemented after the original OLID dataset was enhanced with samples from SOLID. Adversarial attacks used were random character removal and synonyms replacement developed by Morris et al. [42]. The synonym replacement was done to every even-indexed word and the character removal to odd-indexed work in a sentence.

In this study, LSTM based sequence model is used for all the three tasks separately i.e. subtask A (offensive language identification), subtask B (categorization of offence), task C (offence target identification). These three tasks were used with three pre-trained embedding i.e. Word2Vec, GloVe and BERT to evaluate relative performance. In Hierarchical multitask learning (HMTL) configuration, BERT pre-trained embedding is used. In this, all the three tasks, being hierarchical, are trained together. In such a multi task configuration, each task is fed the same input whereas the softmax output of the previous task in the hierarchy is concatenated with the embedding of the current task’s input. The joint loss in a hard parameter sharing configuration is the sum of losses in all the three tasks. The same process is repeated with all configurations of the dataset with all possible pre-trained embedding. The train-test split has been done in the ratio of 80:20, and the test data was not changed with any of the adversarial attack strategies.

All the models were trained on Kaggle Kernels, with batch-size set to 32. Reduced learning rate and early stopping callbacks were used to prevent over-fitting. Initial learning rate was 0.001, which was later divided by 10 when there was no change to the loss after 4 epochs.

6. Results and discussion

| Model  | TaskA     | Training Loss | Training Accuracy | Testing Loss | Testing Accuracy |
|--------|-----------|---------------|-------------------|--------------|-----------------|
| Word2Vec | 0.6255    | 67.83         | 0.5443            | 73.30        |
| TaskB  | 0.7776    | 67.13         | 0.6818            | 71.79        |
| TaskC  | 0.8945    | 70.72         | 0.8176            | 71.79        |
| GloVe  | TaskA     | 0.3880        | 82.48             | 0.5424       | 76.32           |
|        | TaskB     | 0.4866        | 80.45             | 0.6740       | 74.28           |
|        | TaskC     | 0.5914        | 77.89             | 0.7950       | 73.75           |
| BERT   | TaskA     | 0.3777        | 83.51             | 0.4579       | 78.59           |
|        | TaskB     | 0.4964        | 80.95             | 0.5694       | 76.74           |
|        | TaskC     | 0.6400        | 76.46             | 0.6987       | 73.53           |
| HMTL   | TaskA     | 0.3782        | 84.13             | 0.4771       | 78.59           |
|        | TaskB     | 0.5153        | 80.66             | 0.6038       | 75.20           |
|        | TaskC     | 0.6425        | 76.28             | 0.7184       | 73.64           |

Several configurations, as explained above, were attempted. Initially, the model was trained on the OLID dataset. The results were mediocre, but it is to be expected due to the size and imbalance of the OLID dataset. The details of the obtained results are described in Table 2.
As mentioned earlier, offensive text is often identified by offensive lexicon. With the SOLID
Table 6: Test results of McNemar’s statistical hypothesis test

| Dataset                               | Subtask | McNemar’s statistic | p value |
|---------------------------------------|---------|---------------------|---------|
| Original dataset without framework    | A       | 344.000             |         |
| Augmented dataset as per framework    | A       | 162.000             | 0.000   |
| Original dataset without framework    | B       | 237.000             |         |
| Augmented dataset as per framework    | B       | 148.000, 0.000      |         |
| Original dataset without framework    | C       | 191.000             |         |
| Augmented dataset as per framework    | C       | 256.000             | 0.000   |

dataset infused to address the class imbalance, the scores shot up nearly 20% for each task, which can be observed from Table 3. We believe the SOLID dataset gave a boost to an already large vocabulary of the model, thus improving the scores.

Next, we further increased the size of the dataset by using “Adversarial Attacks” on the dataset and trained the model on that, and the score can be observed in Table 4. But we can notice that it hardly made any big improvement in the scores. The possible reason became evident from word cloud analysis of the corpus done before and after adversarial training. Adversarial attack possibly augmented the robustness of the model but due to the type of attacks we have chosen, additional lexicons were introduced that were not necessarily markers of offensive language. So, the two kind of balanced each other. Tweets are most of the time spelled wrong, and to reduce the time to send a message, characters are removed from words if they do not contribute to the pronunciation. And with the SOLID dataset also included, the vocabulary was already big enough that replacing it with synonyms didn’t affect the score much. But it was decided to retain the adversarial attack functionality for the desired robustness of the model.

For our final version of the model, we first trained the HMTL model on a “Sentiment Analysis” dataset and retrained it for our dataset. We believe this can improve the word embeddings of the model, and it surely did. Using this strategy, we were able to improve the overall score by 4% for tasks B and C and 1% for task A as seen on Table 5.

If we compare the top scoring work by Ping Liu et al.[27] in SEMEVAL 2019 task 6 on OLID dataset, the highest test accuracies using BERT pre-trained embedding were 86.28 for subtask A, 89.58 for subtask B and 72.77 for subtask C. Using the proposed approach framework, we have been able to achieve 99.27 on subtask A, 94.36 on subtask B and 92.45 on subtask C far exceeding the best performing results.

6.1. Statistical analysis
Statistical hypothesis test has been performed to prove the relative efficacy of the framework in this case study. The popular approach to use Paired Student’s t-Test has the weakness that each evaluation of the model is not independent resulting in biased t-Test results. The MLxtend library provides an implementation of the approach [60] by Thomas Dietterich but this is not suitable for a deep learning model where evaluating multiple times may not be feasible. Hence, as suggested by Thomas Dietterich[60], McNemar’s non-parametric statistical hypothesis test has been used by making a contingency table to compare the two models i.e. original model where this framework is not used and final model where the proposed framework is used. McNemar’s test essentially checks if the disagreements on the contingency tables for the two classifiers match. The test results are tabulated in Table 6. In this case, there are different proportions of error (reject null hypothesis H0) and the p-value is about 0.000 which is less than 0.05 for 5% significance, suggesting that any observed difference between the algorithms is probably real.
7. Conclusion and future work
Disinformation is a reality of modern life and is rampant on social media. Few of the key challenges faced by disinformation research are the lack of sufficient labeled dataset, subtlety of the language used, adversarial attacks and implicit or explicit hierarchy in the sub-categories of a disinformation type. A technique such as multi-task learning has been successfully used by researchers to address the lack of large dataset by simultaneous learning through related tasks. Even this approach faces the bottleneck that available dataset are not labeled for all related tasks. While the linguistic approach is good when sufficient labeled data is available, the above factors make it unviable.

The work presented in this paper proposes an approach framework that can work around the above difficulties and is shown to better the best performance achieved so far by a large amount. As a case study, this work adopts the problem of offensive language detection and documents the approach in a step by step fashion with requisite details.

This approach framework can be easily replicated for other types of disinformation dataset having implicit or explicit hierarchies. As discussed previously, most disinformation types have class hierarchies and the approach framework documented here is equally applicable for each. The logical next step in this work will be to apply this framework for disinformation having explicit class hierarchies such as persuasion detection in multi-modal memes, propaganda detection along with identification of propaganda technique used etc. We also plan to improve on the hierarchical multi-task part of the framework by exploring hyper parameters for hard parameter sharing and other methods of joint loss optimisation.

References
[1] Su Q, Wan M, Liu X and Huang C R 2020 Natural Language Processing Research 1 1–13
[2] Ruder S, Peters M E, Swayamdipta S and Wolf T 2019 Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials pp 15–18
[3] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H and He Q 2020 Proceedings of the IEEE 109 43–76
[4] Wang Z, Dai Z, Póczos B and Carbonell J 2019 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp 11293–11302
[5] Standley T, Zamir A, Chen D, Guibas L, Malik J and Savarese S 2020 International Conference on Machine Learning (PMLR) pp 9120–9132
[6] Day O and Khoshgoftaar T M 2017 Journal of Big Data 4 1–42
[7] Pennington J, Socher R and Manning C D 2014 Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) pp 1532–1543
[8] Bojanowski P, Grave E, Joulin A and Mikolov T 2017 Transactions of the Association for Computational Linguistics 5 135–146
[9] Devlin J, Chang M W, Lee K and Toutanova K 2018 arXiv preprint arXiv:1810.04805
[10] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L and Stoyanov V 2019 arXiv preprint arXiv:1907.11692
[11] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R R and Le Q V 2019 Advances in neural information processing systems 32
[12] Howard J and Ruder S 2018 arXiv preprint arXiv:1801.06146
[13] Cer D, Yang Y, Kong S y, Hua N, Limtiaco N, John R S, Constant N, Guajardo-Céspedes M, Yuan S, Tar C et al. 2018 arXiv preprint arXiv:1803.11175
[14] Kumar A, Ekbal A, Kawalira D and Kurohashi S 2019 2019 International Joint Conference on Neural Networks (IJCNN) (IEEE) pp 1–8
[15] Akhtar M S, Chauhan D S, Ghosal D, Poria S, Ekbal A and Bhattacharyya P 2019 arXiv preprint arXiv:1905.05812
[16] Li Y and Caragea C 2019 Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) pp 6299–6305
[17] Hazarika D, Krishnamurthy G, Poria S and Zimmermann R
[18] Wu L, Rao Y, Jin H, Nazir A and Sun L 2019 arXiv preprint arXiv:1909.01720
[19] Li Q, Zhang Q and Si L 2019 Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics pp 1173–1179
[20] Kochkina E, Liakata M and Zubiaga A 2018 arXiv preprint arXiv:1806.03713
[21] Kapil P and Ekbal A 2020 Knowledge-Based Systems 210 106458
[22] Vaidya A, Mai F and Ning Y 2020 Proceedings of the International AAAI Conference on Web and Social Media vol 14 pp 683–693
[23] Dai W, Yu T, Liu Z and Fung P 2020 arXiv preprint arXiv:2004.13432
[24] Mozafari M, Farahbaksh R and Crespi N 2019 International Conference on Complex Networks and Their Applications (Springer) pp 928–940
[25] Slovikovskaya V 2019 arXiv preprint arXiv:1910.14353
[26] d'Sa A G, Illina I and Fohr D 2020 2020 International Multi-Conference on: “Organization of Knowledge and Advanced Technologies” (OCTA) (IEEE) pp 1–5
[27] Liu P, Li W and Jozu L 2019 Proceedings of the 13th international workshop on semantic evaluation pp 87–91
[28] Rizoiu M A, Wang T, Ferraro G and Suominen H 2019 arXiv preprint arXiv:1906.03829
[29] Ekman P 1992 Cognition & emotion 6 169–200
[30] Plutchik R 2001 American Scientist 89 344–350
[31] Da San Martino G, Barrón-Cedeno A, Wachsmuth H, Petrov R and Nakov P 2020 Proceedings of the Fourteenth Workshop on Semantic Evaluation pp 1377–1414
[32] Dimitrov D, Ali B B, Shaar S, Alam F, Silvestri F, Firooz H, Nakov P and Martino G D S 2021 arXiv preprint arXiv:2105.09284
[33] Vaidya A, Mai F and Ning Y 2020 Proceedings of the International AAAI Conference on Web and Social Media vol 14 pp 683–693
[34] Rizoiu M A, Wang T, Ferraro G and Suominen H 2019 arXiv preprint arXiv:1906.03829
[35] Ekman P 1992 Cognition & emotion 6 169–200
[36] Plutchik R 2001 American Scientist 89 344–350
[37] Da San Martino G, Barrón-Cedeno A, Wachsmuth H, Petrov R and Nakov P 2020 Proceedings of the Fourteenth Workshop on Semantic Evaluation pp 1377–1414
[38] Dimitrov D, Ali B B, Shaar S, Alam F, Silvestri F, Firooz H, Nakov P and Martino G D S 2021 arXiv preprint arXiv:2105.09284
[39] Derczynski L, Bontcheva K, Lukasik M, Declerck T, Scharl A, Georgiev G, Osenova P, Lobo T P, Kolliakou A, Stewart R et al. 2015 Proceedings of the Extended Semantic Web Conference EU Project Networking session (ESWC-PN)
[40] Zweig A and Weisshull D 2013 Proceedings of the ICML (Citeseer)
[41] Naik A, Charuvaka A and Rangwala H 2013 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (IEEE) pp 390–397
[42] Solomon J and Charette F 2015 arXiv preprint arXiv:1502.05466
[43] Gao L, Huang D, Guo Y and Wang Y 2019 Proceedings of the 27th ACM International Conference on Multimedia pp 1340–1348
[44] Xu L, Fan H and Xiang J 2019 2019 IEEE International Conference on Image Processing (ICIP) (IEEE) pp 3861–3865
[45] Song W, Song Z, Liu L and Fu R 2020 Proceedings of the 29th International Joint Conference on Artifical Intelligence (IJCAI) pp 3875–3881
[46] Yang L, Ng T L J, Smyth B and Dong R 2020 Proceedings of The Web Conference 2020 pp 441–451
[47] Miyato T, Dai A M and Goodfellow I 2016 arXiv preprint arXiv:1605.07725
[48] Morris J, Lifland E, Yoo J Y, Grigsby J, Jin D and Qi Y 2020 Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations pp 119–126
[49] Whitfield D 2021 arXiv preprint arXiv:2014.10658
[50] Fortuna P and Nunes S 2018 ACM Computing Surveys (CSUR) 51 1–30
[51] Waseem Z, Davidson T, Warmsley D and Weber I 2017 arXiv preprint arXiv:1705.09899
[52] Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N and Kumar R 2019 arXiv preprint arXiv:1902.09666
[53] Vosoughi S, Roy D and Aral S 2018 Science 359 1146–1151
[54] Pedersen T 2020 arXiv preprint arXiv:2007.12949
[55] Dadvar M, Trieschnigg D, Ordelman R and de Jong F 2013 European Conference on Information Retrieval (Springer) pp 693–696
[56] Wiegand M, Siegel M and Ruppenhofer J 2018
[57] Kumar R, Öjha A K, Malmasi S and Zampieri M 2018 Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018) pp 1–11
[58] Elmadany A, Zhang C, Abdul-Mageed M and Hashemi A 2020 arXiv preprint arXiv:2006.01266
[59] Vyshnav M, Kumar S and Soman K 2020 arXiv preprint arXiv:2001.03131
[60] Zhang Z, Robinson D and Tepper J 2018 European semantic web conference (Springer) pp 745–760
[61] Goldberg Y and Levy O 2014 arXiv preprint arXiv:1402.3722
[62] Caselli T, Basile V, Mitrovic J, Kartozyia I and Granitzer M 2020 Proceedings of The 12th Language Resources and Evaluation Conference pp 6193–6202
[63] Rosenthal S, Atanasova P, Karadzhov G, Zampieri M and Nakov P 2020 arXiv preprint arXiv:2004.14454
[64] Gupta S and Waseem Z 2017 A comparative study of embeddings methods for hate speech detection from tweets
[65] Malmasi S and Zampieri M 2018 Journal of Experimental & Theoretical Artificial Intelligence 30 187–202
[66] Dietterich T G 1998 Neural computation 10 1895–1923