Emoji Powered Capsule Network to Detect Type and Target of Offensive Posts in Social Media

Hansi Hettiarachchi[1], **Tharindu Ranasinghe**[2]
[1]Department of Computer Science and Engineering,, University of Moratuwa, Sri Lanka
[2]Research Group in Computational Linguistics, University of Wolverhampton, UK
hansi.11@cse.mrt.ac.lk
t.d.ranasinghehettiarachchige@wlv.ac.uk

Abstract

This paper describes a novel research approach to detect type and target of offensive posts in social media using a capsule network. The input to the network was character embeddings combined with emoji embeddings. The approach was evaluated on all the subtasks in SemEval-2019 Task 6: OffensEval: Identifying and Categorizing Offensive Language in Social Media. The evaluation also showed that even though the capsule networks have not been used commonly in NLP tasks, they can outperform existing state of the art solutions for offensive language detection in social media.

1 Introduction

Social media has become a normal medium of communication for people these days as it provides the convenience of sending messages fast from a variety of devices. Unfortunately, social networks also provide the means for distributing abusive and aggressive content. Given the amount of information generated every day on social media, it is not possible for humans to identify and remove such messages manually, instead it is necessary to employ automatic methods. Recently, many shared tasks have been introduced to encourage the development of methods capable of classifying messages from social media as offensive. As an example, the First Workshop on Trolling, Aggression and Cyberbullying has organised the First Shared Task on Aggression Identification to classify messages from Facebook and Twitter into three categories Overtly Aggressive (OAG), Covertly Aggressive (CAG) and Non-aggressive (NAG) (Kumar et al., 2018). The task was organised for English and Hindi.

Recently, more complete dataset covering different aspects of offensive identification was released for the shared task in SemEval-2019 Task 6: OffensEval: Identifying and Categorizing Offensive Language in Social Media. The task was not only to identify offensive messages in social media. The participants had to categorize the offensive language and also had to identify the targeted audience (Zampieri et al., 2019). We used this dataset to experiment our novel architecture since it covers more aspects in offensive language detection in social media. More details about the tasks and the dataset will be discussed in Section 2.

People from all over the world uses social media. Therefore, a random sample of social media messages can be written in several languages. As a result, systems that detect offensive posts without relying too much on language dependent features would be valuable in real life scenarios. In the offensive language identification shared tasks too, most researches have worked on systems that rely on word/character embeddings rather than linguistics features. As an example Galery and Charitos (2018) has taken an approach to feed fasttext (Mikolov et al., 2018) character embeddings to a Gated Recurrent Neural Network architecture (Chung et al., 2014). As the system doesn't rely on linguistic features, it is easily portable between Hindi and English. These type of architectures can be easily implemented for other languages once the data for training is available.

Most approaches in shared tasks are based on word/character embeddings feeding to a neural network (Kumar et al., 2018). Most of these architectures use max pooling or successive convolutional layers that reduce spacial size of the data flowing through the network and therefore increase the view of higher layers neurons, allowing them to detect higher order features in a larger

Proceedings of Recent Advances in Natural Language Processing, pages 474–480,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_056

region of the input embeddings. However recent introduction of capsule networks shows that while pooling works better in most of the scenarios, it nonetheless is losing valuable information (Hinton et al., 2018). The solution that has been brought forward is Capsule Networks. How it overcomes the weaknesses in max pooling layer will be discussed in Section 3.

Since the Capsule Networks are very new to the field, they have not been used much in NLP tasks. However, their good performance in image classification tasks motivated us to use them in offensive language detection tasks too. To the best of our knowledge, no prior work has been explored in offensive language identification with Capsule Networks. Also, it might be important to explore how the Capsule Networks performs in NLP domain. Additionally we analyzed that most of the social media posts contain not only text but emojis too, which can be a contributing factor for offense. Therefore we propose a method to incorporate emoji knowledge to the Capsule Network architecture. Generally, this paper proposes a Capsule Network architecture with emoji information to detect offensive posts in social media. The rest of the paper is organised as follow. Section 2 would briefly describe the tasks and the dataset. Section 3 would describe the capsule network architecture we used and how we integrated emoji information to the architecture. After that we evaluate the system comparing with the architectures provided in Zampieri et al. (2019). Finally, the conclusions are presented.

2 Dataset and Task Description

Dataset that we used was released for the Task 6 in SemEval-2019 : OffensEval: Identifying and Categorizing Offensive Language in Social Media. It has been collected from Twitter using its API and searching for keywords and constructions that are often included in offensive messages, such as she is, to:BreitBartNews, gun control etc (Zampieri et al., 2019).

There were three tasks associated with the shared task.

- *Subtask A: Offensive language Detection* : Goal of the task was to discriminate between the following types of tweets:
  - **Not Offensive (NOT)** : Posts that do not contain offense
  - **Offensive (OFF)**: Posts containing any form of non-acceptable language. These posts can include insults, threats swear words etc. (Zampieri et al., 2019)
- *Subtask B: Categorization of Offensive Language :* Task's goal was to categorize the type of offense.
  - **Targeted Insult (TIN)** : Posts that contain targeted insults and threats.
  - **Untargeted (UNT)** : Posts containing non targeted insults or threats.
- *Subtask C: Offensive Language Target Identification :* Goal of the task was to categorize the targets of insults/threats.
  - **Individual (IND)** : Insults that target individuals.
  - **Group (GRP)** : Insults that target a group of people.
  - **Other (OTH)** : The target does not belong to any category mentioned above.

Few examples from the training set is shown in Table 1.

As you can see, the nature of the three tasks are different and it would interesting to explore how one architecture can be used to capitalise all of the three tasks.

3 Research Approach

We first describe the existing approaches mentioned in Zampieri et al. (2019). Then we will describe the proposed capsule network architecture.

3.1 Existing Approaches

There are three approaches considered in Zampieri et al. (2019) which will be described in the following list. We describe them briefly in this sub section before introducing the capsule network architecture.

1. **SVM** - A linear SVM trained on word unigrams. SVMs have achieved state-of-the-art results for many text classification tasks (Zampieri et al., 2018).

2. **BiLSTM** - The model consists of (i) an input embedding layer,(ii) a bidirectional LSTM layer (Schuster and Paliwal, 1997), and (iii) an average pooling layer of input features. The concatenation of the LSTM layer and

id	tweet	a	b	c
44209	@USER @USER what a baby! URL	NOT	NULL	NULL
97670	@USER Liberals are all Kookoo !!!	OFF	TIN	OTH
74831	@USER Trump kicks dem butt - its so fun.	OFF	TIN	IND
17259	IM FREEEEE!!!! WORST EXPERIENCE OF MY FUCKING LIFE	OFF	UNT	NULL

Table 1: Example rows from the dataset

the average pooling layer is further passed through a dense layer, whose output is ultimately passed through a softmax to produce the final prediction. The model is adapted from a pre-existing model for sentiment analysis (Rasooli et al., 2017).

3. **CNN** - A convolutional neural network based on the model proposed in Kim (2014). It consists of an (i) an input embedding layer, (ii) a convolutional layer (Collobert et al., 2011) and (iii) a max pooling layer (Collobert et al., 2011) of input features. The output of the max pooling layer is further passed through a dropout (Srivastava et al., 2014) and softmax output.

Both BiLSTM and CNN architectures above have pooling layers which is a very primitive type of routing mechanism. The most active features in a local pool is routed to the higher layer and the higher-level detectors don't have an impact in the routing. However in the Capsule Network, only those features that agree with high-level detectors are routed. It has a superior dynamic routing mechanism. With this advantage we propose a novel capsule network architecture for aggression detection which will be described in the next section.

3.2 Proposed Architecture

The proposed architecture is depicted in Figure 1. The architecture consists of four layers.

1. *Embedding Layer* - We represent every text x_i , as a sequence of one-hot encoding of its words, $x_i = (w_1, w_2, ...w_n)$ of length n, which is the maximum length of the all of the texts in the training set, with zero padding. Such a sequence becomes the input to the embedding layer. Most of the words exist in social media texts are not proper words. If we used word embeddings to initialize the embedding matrix in the embedding layer, most of the words that are fed will be out-of-vocabulary words. Therefore we used character embeddings (Mikolov et al., 2018) as it provides embeddings for misspelling words and new words. Also character embeddings handle infrequent words better than word2vec embedding as later one suffers from lack of enough training opportunity for those rare words. We used fasttext embeddings pre trained on Common Crawl (Mikolov et al., 2018). Using the model we represented each word as a vector with a size of 300 values. The embedding layer is improved more with emoji information, which will be described in Section 3.3.

2. *Feature Extraction Layer* - We used this layer to extract long term temporal dependencies within the text. We experimented both LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Chung et al., 2014) for this layer. Due to the fact that we had a small number of training examples, GRUs performed better than LSTMs, capitalising on GRU's ability to exhibit better performance on smaller datasets. For the final architecture we used a bi directional GRU layer with 50 time steps, each getting initialised with glorot normal initialiser.

3. *Capsule Layer* - The Capsule layer we used is primarily composed of two sub-layers *Primary Capsule Layer* and *Convolutional Capsule Layer*.

   (a) *Primary Capsule Layer* - The primary capsule layer is supposed to capture the instantiated parameters of the inputs, for example, in case of texts, local order of words and their semantic representation is captured with the primary capsule layer.

   (b) *Convolutional Capsule Layer* - The convolutional capsule layer outputs a lo-

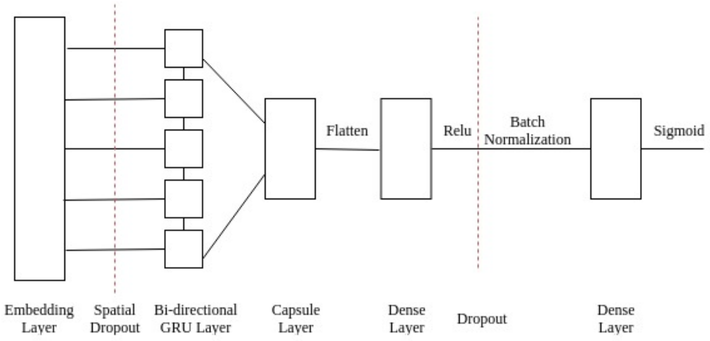


Figure 1: Capsule Network

cal grid of vectors to capsules in earlier layers using different transformation matrices for each capsule. This is trained using dynamic routing algorithm described in Sabour et al. (2017) that overlooks words that are not important or unrelated in the text, like stopwords and name mentions which are common in social media texts.

4. *Dense Layers* - Output of the Capsule layer is flattened and then fed in to two dense layers. First dense layer had 100 units and was activated with relu function. After applying batch normalization to the output of first dense layer, it was fed in to the second dense layer with 1 unit and and sigmoid activation.

Apart from the major sections in the architecture described above, we used a spatial dropout (Tompson et al., 2015) between the embedding layer and the feature extraction layer and a dropout (Srivastava et al., 2014) between the two dense layers to minimize over fitting of the network. The implementation was done using Keras (Chollet et al., 2015) and Python[1].

The next section would describe how we integrated emoji knowledge to this architecture.

3.3 Integrating Emojis

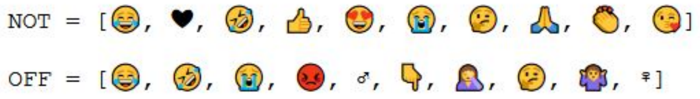
Emojis are ideograms which are used with text to visually complement its meaning. In present, emojis are widely used by social media. A global analysis done on Twitter has been found that 19.6% of tweets contain emojis. Further it stated, emojis are used by 37.6% of users (Ljubesic and Fiser, 2016). A research conducted by (Barbieri et al., 2017) has been showed that there is an unique and important relation between sequences of words and emojis. When analyze the top 10 emojis belong to both categories; Offensive and Not Offensive, in the selected dataset, it also shows a clear distinction of emojis corresponding to its category as shown in Figure 2. Due to the extensive usage of emojis in social media and the relationship lie between emojis and text, integration of emojis can be used to improve the social media offensive language detection.

Since the proposed architecture is based on embeddings, we decided to integrate emojis also using the embeddings. But most of the available pre-trained word embedding sets include few or no emoji representations. Therefore in addition to the character embeddings, separate embedding set; emoji2vec (Eisner et al., 2016) was chosen for emojis. Emoji2vec consists of pre-trained embeddings for all Unicode emojis using their descriptions in the Unicode emoji standard. This maps emojis into 300-dimensional space similar to other available word embeddings; word2vec, glove, etc. to make the integration easy with word vectors. Emoji2vec embeddings were evaluated based on sentiment analysis on tweets and it showed word2vec with emoji embeddings advances the classification accuracy while proving that the emoji2vec embeddings are useful in social natural language processing tasks.

Following (Eisner et al., 2016), there were two pre-trained emoji2vec models[2]. One model is based on the sum of vectors corresponds to the words found in phrases which describe the emojis. As an extended version of it, other model feeds the actual word embeddings to an LSTM layer. We

[1] The code is available on "https://github.com/TharinduDR/Aggression-Identification"

[2] https://github.com/uclmr/emoji2vec



```
NOT = [😂, ♥, 🤣, 👍, 😍, 😭, 🤔, 🙏, 👏, 😘]
OFF = [😂, 🤣, 😭, 😡, ♂, 👇, 🤦, 🤔, 🤷, ♀]
```

Figure 2: Top 10 emojis belong to Not Offensive (NOT) and Offensive (OFF) posts, Task 6 Dataset, SemEval-2019

Model	NOT			OFF			Weighted Average			
	P	R	F1	P	R	F1	P	R	F1	F1 Macro
SVM	0.80	0.92	0.86	0.66	0.43	0.52	0.76	0.78	0.76	0.69
BiLSTM	0.83	0.95	0.89	0.81	0.48	0.60	0.82	0.82	0.81	0.75
CNN	0.87	0.93	0.90	0.78	0.63	0.70	0.82	0.82	0.81	0.80
CapsuleNet †	0.88	0.93	0.91	0.82	0.64	0.71	0.83	0.82	0.82	**0.81**
All NOT	-	0.00	0.00	0.72	1.00	0.84	0.52	0.72	0.	0.42
All OFF	0.28	1.00	0.44	-	0.00	0.00	0.08	0.28	0.12	0.22

Table 2: Results for offensive language detection. We report Precision (P), Recall (R), and F1 for each model/baseline on all classes (NOT, OFF), and weighted averages. Macro-F1 is also listed (best in bold). † denotes the capsule net architecture integrated with emoji embeddings.

used 300 dimensional embedding spaces generated by both models; sum based and LSTM based for this experiment.

Two approaches were used to integrate emoji embeddings with the above mentioned architecture as follows:

1. *300 dimensional embedding layer* - Shared same 300 dimensional vector space for both word and emoji embeddings.

2. *600 dimensional embedding layer* - Used concatenation layer and resulted 600 dimensional vector space by both word and emoji embeddings.

Among the experiments we conducted using both emoji2vec models and integration approaches, combination of sum based emoji embeddings with 600 dimensional embedding layer and LSTM based emoji embeddings with 300 dimensional embedding layer resulted improvements compared to word embeddings only approaches. More details on experiment results are mentioned in Section 4.

3.4 Training

The network was trained on the training dataset provided for SemEval-2019 Task 6: OffensEval: Identifying and Categorizing Offensive Language in Social Media. It was trained using adam optimiser (Kingma and Ba, 2015), with a reduced learning rate once learning stagnates. Also the parameters of the network was optimized using five fold cross validation.

4 Evaluation

The capsule network architecture we proposed above was evaluated using the testing set provided for each of the subtask in SemEval-2019 Task 6: OffensEval: Identifying and Categorizing Offensive Language in Social Media.

4.1 Offensive Language Detection

The performance on identifying offensive (OFF) and non-offensive (NOT) posts is reported in Table 2. The Capsule Network we proposed outperforms the RNN model, achieving a macro-F1 score of 0.81.

4.2 Categorization of Offensive Language

The results for the offensive language categorization is shown in Table 3. In this subtask too Capsule Network architecture outperforms all the other models having a macro F1 score of 0.71.

4.3 Offensive Language Target Identification

The results for the offensive language target identification is shown in Table 4. Capsule Network architecture outperforms all the other models having a macro F1 score of 0.49.

Model	TIN			UNT			Weighted Average			
	P	R	F1	P	R	F1	P	R	F1	F1 Macro
SVM	0.91	0.99	0.95	0.67	0.22	0.33	0.88	0.90	0.88	0.64
BiLSTM	0.95	0.83	0.88	0.32	0.63	0.42	0.88	0.81	0.83	0.66
CNN	0.94	0.90	0.92	0.32	0.63	0.42	0.88	0.86	0.87	0.69
CapsuleNet †	0.96	0.94	0.94	0.33	0.67	0.44	0.90	0.88	0.87	**0.71**
All TIN	0.89	1.00	0.94	-	0.00	0.00	0.79	0.89	0.83	0.47
All UNT	-	0.00	0.00	.11	1.00	0.20	.01	0.11	0.02	0.10

Table 3: Results for offensive language categorization. We report Precision (P), Recall (R), and F1 for each model/baseline on all classes (TIN, UNT), and weighted averages. Macro-F1 is also listed (best in bold). † denotes the capsule net architecture integrated with emoji embeddings.

.

Model	GRP			IND			OTH			Weighted Average			
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	F1 Macro
SVM	0.66	0.50	0.57	0.61	0.92	0.73	0.33	0.03	0.05	0.58	0.62	0.56	0.45
BiLSTM	0.62	0.69	0.65	0.68	0.86	0.76	0.00	0.00	0.00	0.55	0.66	0.60	0.47
CNN	0.75	0.60	0.67	0.63	0.94	0.75	0.00	0.00	0.00	0.57	0.66	0.60	0.47
Capsule Net †	0.78	0.65	0.70	0.64	0.95	0.78	0.02	0.03	0.02	0.59	0.68	0.62	**0.49**
All GRP	0.37	1.00	0.54	-	0.00	0.00	-	0.00	0.00	0.13	0.37	0.20	0.18
All IND	-	0.00	0.00	0.47	1.00	0.64	-	0.00	0.00	0.22	0.47	0.30	0.21
All OTH	-	0.00	0.00	-	0.00	0.00	0.16	1.00	0.28	0.03	0.16	0.05	0.09

Table 4: Results for offense target identification. We report Precision (P), Recall (R), and F1 for each model/baseline on all classes (GRP, IND, OTH), and weighted averages. Macro-F1 is also listed (best in bold). † denotes the capsule net architecture integrated with emoji embeddings.

As shown in Table 2, 3 and 4 capsule network architecture outperformed all the other models in all sub tasks. It is worth noticing that the unbalanced nature of the dataset did not affect the performance of the capsule network architecture. Also eventhough the capsule layer is seemingly complex, results show that it does not need a large training set to optimize its parameters.

We did not fine tune the model analyzing data in this dataset since we wanted a general model capable of identifying offense. Hence, we did not compare our results with the final results of the shared task.

5 Conclusion

We have presented a novel capsule network architecture to detect type and target of offensive posts in social media. Also we propose a method to incorporate emoji knowledge to the architecture. Our approach was able to improve on the baseline system presented at SemEval-2019 Task 6: OffensEval: Identifying and Categorizing Offensive Language in Social Media. Importantly our system does not rely on language dependent features so that it is portable for any other language too.

The main conclusion of the paper is that even though the capsule networks are not widely used in NLP domain, they can achieve state of the art results. Also with the shown way of integrating emoji information to the network, results can improve.

In the future we hope to implement a multi purpose capsule network architecture for several tasks in NLP domain such as spam detection, gender identification etc. We hope to further explore capsule network architectures in various NLP tasks.

References

Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are emojis predictable? *Arxiv* .

François Chollet et al. 2015. Keras. https://keras.io.

Junyoung Chung, aglar Gülehre, Kyunghyun Cho, and

Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their descriptions. *Arxiv* .

Thiago Galery and Efstathios Charitos. 2018. Aggression identification and multi lingual word embeddings. In *TRAC@COLING 2018*.

Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix capsules with em routing. In *ICLR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9:1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *TRAC@COLING 2018*.

Nikola Ljubesic and Darja Fiser. 2016. A global analysis of emoji usage. In *WAC@ACL*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Mohammad Sadegh Rasooli, Noura Farra, Axinia Radeva, Tao Yu, and Kathleen McKeown. 2017. Cross-lingual sentiment transfer with limited resources. *Machine Translation* 32:143–165.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. *ArXiv* abs/1710.09829.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing* 45:2673–2681.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient object localization using convolutional networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pages 648–656.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardzic, Nikola Ljubesic, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language identification and morphosyntactic tagging: The second vardial evaluation campaign. In *VarDial@COLING 2018*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *Proceedings of NAACL*.