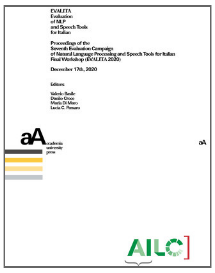

EVALITA
Evaluation
of NLP
and Speech Tools
for Italian

Proceedings of the
Seventh Evaluation Campaign
of Natural Language Processing and Speech Tools for Italian
Final Workshop (EVALITA 2020)

December 17th, 2020

Editors:

Valerio Basile
Danilo Croce
Maria Di Maro
Lucia C. Passaro

Valerio Basile, Danilo Croce, Maria Maro and Lucia C. Passaro (dir.)

EVALITA Evaluation of NLP and Speech Tools for Italian - December 17th, 2020
Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian Final Workshop

Accademia University Press

# fabsam @ AMI: A Convolutional Neural Network Approach

Samuel Fabrizi

DOI: 10.4000/books.aaccademia.6782
Publisher: Accademia University Press
Place of publication: Torino
Year of publication: 2020
Published on OpenEdition Books: 11 May 2021
Serie: Collana dell'Associazione Italiana di Linguistica Computazionale
Electronic ISBN: 9791280136329

**Electronic reference**
FABRIZI, Samuel. *fabsam @ AMI: A Convolutional Neural Network Approach* In: *EVALITA Evaluation of NLP and Speech Tools for Italian - December 17th, 2020: Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian Final Workshop* [online]. Torino: Accademia University Press, 2020 (generated 17 mai 2021). Available on the Internet: <http://books.openedition.org/aaccademia/6782>. ISBN: 9791280136329. DOI: https://doi.org/10.4000/books.aaccademia.6782.

# fabsam @ AMI: A Convolutional Neural Network Approach

**Samuel Fabrizi**
University of Pisa, Italy
s.fabrizi1@studenti.unipi.it

## Abstract

The presence of misogynistic contents is one of the most crucial problems of social networks. In this paper we present our system for misogyny identification on Twitter. Our approach is based on a convolutional neural network that exploits pretrained word embeddings. We also experimented a comparison among different architectures to understand the effectiveness of our method. The paper also described our submissions to both subtasks A and B to Automatic Misogyny Identification competition at Evalita 2020.

## 1 Introduction

The paper describes our submission to the Automatic Misogyny Identification task at Evalita 2020 (Fersini et al., 2020; Basile et al., 2020). This competition is divided into two subtasks:

- **Subtask A** Misogyny and Aggressive Behaviour Identification: identify if a text is misogynous or not, and, in case of misogyny, if it expresses an aggressive attitude.
- **Subtask B** Unbiased Misogyny Identification: discriminate misogynistic contents from the non-misogynistic ones, while guaranteeing the fairness of the model (in terms of unintended bias) on a synthetic dataset (Nozza et al., 2019).

We proposed a convolutional based approach to recognize misogynistic sentences. We grounded our work over a robust model selection technique. In order to confirm our approach we developed other architectures based on state of art models to make a systematic comparison.
Our work is organized as follows. Section 2 briefly describes related work on the proposed task. Section 3 describes our architectures. Section 4 introduces our method. In particular, it describes our approach for model selection and assessment. Section 5 presents the official results obtained in the AMI competition. Section 6 concludes this work.

## 2 Related Work

The misogyny identification and classification approaches are very recent (Anzovino et al., 2018). In the last few years there was an increasing number of research on this field. The majority of them have concentrated especially on abusive and aggressive language detection. This form of hate speech task has been proposed in different organized shared tasks: IberEval 2018 (Fersini et al., 2018), Evalita 2018 (Fersini et al., 2018) and later at SemEval 2019 (Basile et al., 2019). Most of the state-of-art approaches to misogyny detection were described as system reports for these shared tasks.
Finally, it is important to mention that different deep learning approaches have been proposed (Badjatiya et al., 2017). In this paper we extend the use of convolutional layers for word based feature extraction.

## 3 Description of the system

In this section we describe our approach that exploits the intuition of extracting dependencies among words as features from tweets. We also made an analysis about other architectures and we compare them with ours in order to understand strength and weakness of our architecture. Our method consists of the following steps:

- normalization of the datasets;
- use an effective word embedding representation;

- define different state of art architectures to compare them with our model.

### 3.1 Data Preprocessing and Word Embeddings

Out-of-vocabulary words are one of the most important issues with the use of word embedding, especially in the context of social networks in which colloquial language is widespread. In order to normalize tweets, we pre-processed them using tools from *ekphrasis* (Baziotis et al., 2017).

First of all we removed punctuation and separated sentences into words. Then we applied the normalization process. This process involves, for example, allcaps annotation ('ABC' becomes 'allcaps abc allcaps'), elongated words normalization ('vaaaaai' becomes 'elongated vai elongated') and emoticons transformation. We manually carried out translations of these keywords to adapt annotations to the Italian language.

We experimented different word embedding pretrained model. After a sequence of considerations we chose the word embeddings presented in Cimino et al. (2018) trained on 46 million Italian tweets. It is a word2vec based model and it encodes each word in a 128-size vector.

### 3.2 Our model

The model used for the AMI competition is represented in Figure 1.

Given a tweet, we firstly apply the pre-processing described in Section 3.1 to normalize and transform it into a sequence of words. Then this sequence is mapped into a fixed real vector domain by the embedding layer.

The embedding layer passes an input feature space to three 1D Convolutional layers. Each of those uses 150 filters and a stride of 1 but different kernel sizes of 1, 2, 4 respectively. These layers are the most interesting ones. Each layer can indeed be seen as extractors of n-gram features where n is equal to the kernel size (Kim, 2014). As explained in Section 4.1 we search for the best hyperparameters of these layers in model selection phase.

Outputs from CNN layers are down-sampled by a GlobalMaxPooling1D layer and then they are concatenated into a single sequence.

The last two layers are dense layers with tanh and softmax activation functions respectively. The final softmax layer maps the sequence received as input to a probability distribution over all possible classes.

This model was trained for 15 epochs using a batch size of 128.

## 4 Experiments

In the subtask A we split the training set provided into a train set (4250 tweets) and a test set (750 tweets). This internal test set was used only to evaluate our final model. In subtask B we merged raw and synthetic datasets and separated from each of these two test sets.

As explained in Section 3.2 we used as output layer a dense layer with softmax activation function. In order to obtain three different labels for subtask A, *misogynous* and *aggressiveness* columns were converted into a single one. We also apply one-hot encoding to the integer representation, otherwise a natural ordering between categories may result in poor performance or unexpected results.

The frequency distribution of these labels turns out to be quite unbalanced, as shown in Table 1. Furthermore for each class we have a very small number of training examples. This could have a strong influence on the overfitting of the model. We indeed avoided to use a deep neural network and we preferred to develop a simple architecture in a robust way as recommended in Zhang and Wallace (2015).

| **Class** | **Train set** | **Test set** |
|---|---|---|
| **Non-misogynous** | 2277 | 386 |
| **Non-aggressive** | 484 | 70 |
| **Aggressive** | 1489 | 294 |

Table 1: Subtask A dataset distribution

### 4.1 Model Selection

We decided to apply a robust model selection technique to find the best hyperparameters of our model. We used repeated K-fold cross-validation (Rodriguez et al., 2010).

In subtask A we used the official AMI score as metric. While in the subtask B we decided to use the AUC metric. In both of them we also took into consideration the standard deviation among different runs.

Model selection phase was divided in 2 mainly stages:

- **Stage 1** we validate the best hyperparameters for each different model. We report the hyperparameters ranges in Figure 2. In this

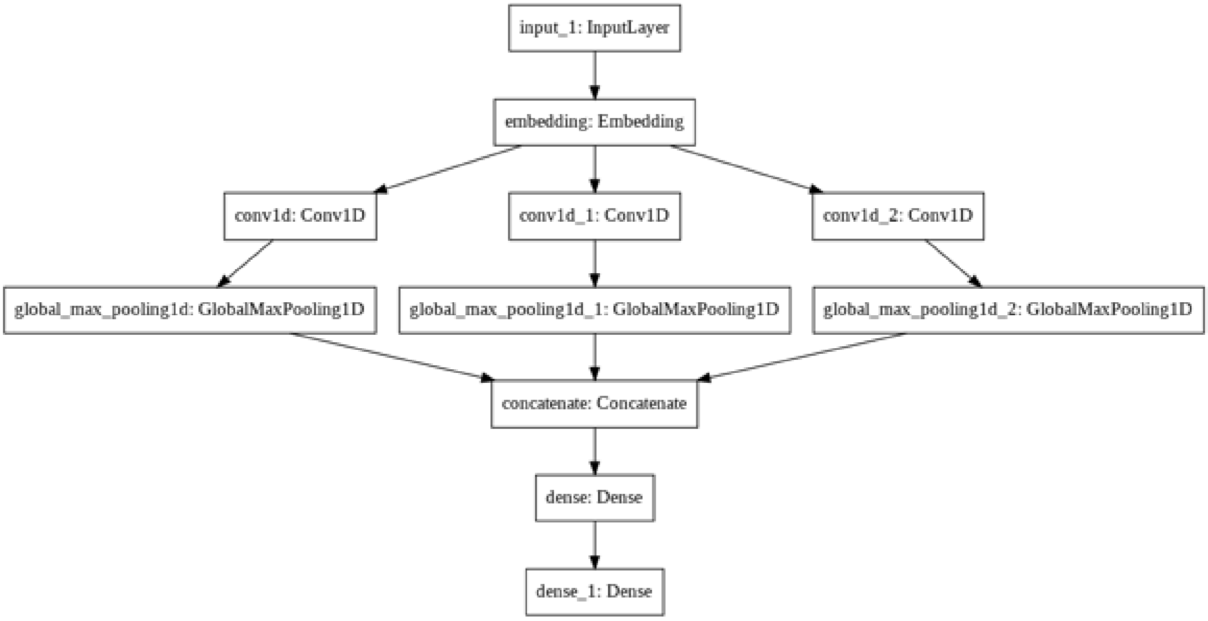

Figure 1: Model architecture

stage we used a repeated 5-fold with 10 repetitions.

- **Stage 2** We chose the most promising models according to score and standard deviation metrics. We applied another repeated 5-fold cross-validation increasing the number of repetitions to 15. Then we chose the best model among them using the same metrics as before.

| Hyperparam | Range |
|---|---|
| **Batch size** | {32, 64, 128} |
| **Filters** | {[100, 100, 100],<br>[150, 150, 150]} |
| **Kernel Sizes** | {[1, 2, 3],<br>[1, 2, 4]} |
| **L2 regularizer** | {0.001, 0.005} |
| **Number dense nodes** | {8, 16} |

Table 2: Hyperparameters ranges

Then we built other architectures to compare them with ours. In the following we list models used for these comparisons:

- Convolution-biGRU Based Deep Neural Network: this architecture allows to capture long-range dependencies from both directions of a sentence;

- Convolutional Based Neural Network: deep neural network based on convolutional layers that tries to extract different features using a greater number of layers. It is an extension of the architecture described in Section 3.2;

- Skipped Convolutional Neural Network (Zhang and Luo, 2018): CNN architecture where each convolutional layer uses “gapped window” to extract features from its input;

In Figure 2 we reported results obtained in stage 2 of the model selection phase in the subtask A. Our model seems to be better in terms of both score and standard deviation compared to the others. Furthermore, it does not have any outliers as other models have.

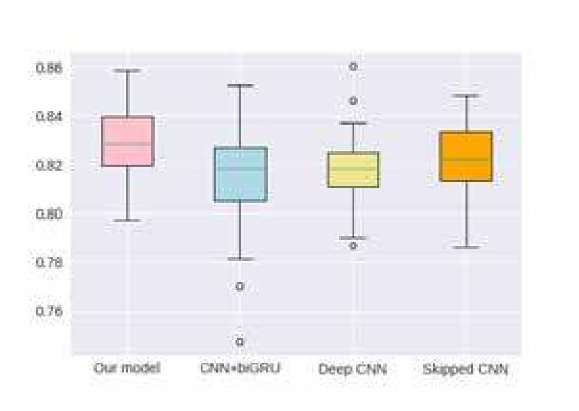

Figure 2: Comparison among different models on subtask A

### 4.2 Model Assessment

As final step we tested our model over the internal test set. The results obtained are reported in Table 3. As expected, the behaviour of our model in this internal test set is in compliance with respect to validation results.
As regards subtask B, we only considered the AUC score and results obtained for both model selection and assessment have proved to be inconclusive.

| Run | Subtask A score |
|---|---|
| **Run 1** | 0.858894 |
| **Run 2** | 0.851679 |
| **Run 3** | 0.8360752 |

Table 3: Results of single runs in internal test set

## 5 Results and discussion

The evaluation was done on both subtask A and B. In the following subsections a discussion of the results obtained in each subtask is provided.

### 5.1 Subtask A

Table 4 reports the official results for the subtask A.

| SubtaskA | u/c | score | teamname |
|---|---|---|---|
| run2 | u | 0.74064 | jigsaw |
| run1 | u | 0.73802 | jigsaw |
| **run1** | **c** | **0.73425** | **fabsam** |
| run1 | u | 0.73135 | YNU_OXZ |
| **run2** | **c** | **0.73091** | **fabsam** |
| run2 | c | 0.71669 | NoPlaceFor.. |
| run2 | u | 0.70145 | YNU_OXZ |
| **run3** | **c** | **0.69482** | **fabsam** |

Table 4: AMI subtask A leaderboard

Both run *fabsam.r.c.run1* and *fabsam.r.c.run1* have outperformed other constrained runs and our best run ranks third in the official leaderboard. This confirms the effectiveness of our approach.
During an error analysis we noticed that our model wrongly classifies short sentences and hate speech sentences referred to men. Nevertheless, in our best run the f1 score for *misogynous* label reaches 0.8038 while the real problem is in the 0.6647 of *aggressiveness* label. This is probably due to the small number of non-aggressive examples used to fit the model. Different results of runs reflect the standard deviation observed during the validation phase. While scores obtained are smaller then model selection results.

### 5.2 Subtask B

In the following we reported our results for the subtask B.

| SubtaskB | u/c | score | teamname |
|---|---|---|---|
| run2 | u | 0.88259 | jigsaw |
| run3 | c | 0.81803 | PoliTeam |
| run1 | c | 0.81369 | PoliTeam |
| **run1** | **c** | **0.70512** | **fabsam** |
| **run2** | **c** | **0.70219** | **fabsam** |
| run2 | c | 0.69395 | PoliTeam |
| run3 | c | 0.69133 | fabsam |
| **run3** | **c** | **0.69133** | **fabsam** |

Table 5: AMI Subtask B leaderboard

We used for subtask B the same model used for the other subtask. We have performed a poor validation approach using as evaluation metric the AUC. We chose to train the model merging raw and synthetic datasets. This choice led to poor performance on unseen datasets. Indeed our model was strongly affected by overfitting when it met identity terms used in training. From an error analysis we noticed that it wrongly classifies lots of sentences from synthetic dataset, while it performs very well on raw dataset.

## 6 Conclusion

The presence of misogynistic contents in social network is a major problem. A crucial work in this direction is the detection and recognition of this type of contents.
We propose a simple architecture based on convolutional layers. From our experiments we understood that capturing long-term dependencies produces an unstable training and poor performance in this type of subtasks. Performances of the model could be increased focusing its approach on model selection. Lastly, it could be very important to take into consideration data augmentation techniques or other sources of data to solve the unbalanced dataset issue.

## References

Maria Anzovino, Elisabetta Fersini, and Paolo Rosso. 2018. Automatic identification and classification of misogynistic language on twitter. In *International Conference on Applications of Natural Language to Information Systems*, pages 57–64. Springer.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August. Association for Computational Linguistics.

Andrea Cimino, Lorenzo De Mattei, and Felice Dell'Orletta. 2018. Multi-task learning in deep neural networks at evalita 2018. *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, pages 86–95.

E Fersini, P Rosso, and M Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. In *Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2018*, volume 2150, pages 214–228. CEUR-WS.

Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2020. Ami @ evalita2020: Automatic misogyny identification. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 08.

Debora Nozza, Claudia Volpetti, and Elisabetta Fersini. 2019. Unintended bias in misogyny detection. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 149–155.

J. D. Rodriguez, A. Perez, and J. A. Lozano. 2010. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):569–575.

Ziqi Zhang and Lei Luo. 2018. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, Accepted, 10.

Ye Zhang and Byron C. Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *CoRR*, abs/1510.03820.