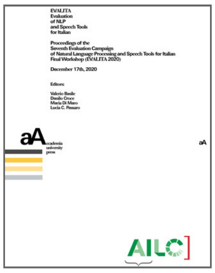

EVALITA
Evaluation
of NLP
and Speech Tools
for Italian

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

aAccademia
university
press

aA

AILC

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

YNU_OXZ @ HaSpeeDe 2 and AMI : XLM-RoBERTa with Ordered Neurons LSTM for classification task at EVALITA 2020

Xiaozhi Ou and Hongling Li

DOI: 10.4000/books.aaccademia.6912
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
OU, Xiaozhi ; LI, Hongling. *YNU_OXZ @ HaSpeeDe 2 and AMI : XLM-RoBERTa with Ordered Neurons LSTM for classification task at EVALITA 2020* 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 May 2021). Available on the Internet: <http://books.openedition.org/aaccademia/6912>. ISBN: 9791280136329. DOI: https://doi.org/10.4000/books.aaccademia.6912.

YNU_OXZ @ HaSpeeDe 2 and AMI : XLM-RoBERTa with Ordered Neurons LSTM for classification task at EVALITA 2020

Xiaozhi Ou
Yunnan University
China
xiaozhiou88@gmail.com

Hongling Li✉
Yunnan University
China
honglingli66@126.com

Abstract

English. This paper describes the system that team YNU_OXZ submitted for EVALITA 2020. We participate in the shared task on Automatic Misogyny Identification (AMI) and Hate Speech Detection (HaSpeeDe 2) at the 7th evaluation campaign EVALITA 2020. For HaSpeeDe 2, we participate in Task A - Hate Speech Detection and submitted two-run results for the news headline test and tweets headline test, respectively. Our submitted run is based on the pre-trained multi-language model XLM-RoBERTa, and input into Convolution Neural Network and K-max Pooling (CNN + K-max Pooling). Then, an Ordered Neurons LSTM (ON-LSTM) is added to the previous representation and submitted to a linear decision function. Regarding the AMI shared task for the automatic identification of misogynous content in the Italian language. We participate in subtask A about Misogyny & Aggressive Behaviour Identification. Our system is similar to the one defined for HaSpeeDe and is based on the pre-trained multi-language model XLM-RoBERTa, an Ordered Neurons LSTM (ON-LSTM), a Capsule Network, and a final classifier.

1 Introduction and Background

People use offensive contents in their social media posts to degrade an individual or religion or other organizations in many respects, the identification of such social media posts is a necessity, a substantial amount of work has been done in languages like English. However, hate speech and offensive language identification in other language scenario is still an area worth exploring. The latest edition of EVALITA (Caselli et al., 2018) hosted the first Hate Speech (HS) detection in Social Media (i.e. HaSpeeDe (Bosco et al., 2018)) task for Italian, the HaSpeeDe 2 (Hate Speech Detection) (Sanguinetti et al., 2020) shared task have been organized within Evalita 2020 [1]. The ultimate goal of HaSpeeDe 2 is to take a step further in the state of the art of HS detection for Italian while also exploring other side phenomena, the extent to which they can be distinguished from HS, and finally whether and how much automatic systems are able to draw such conclusions. For AMI (Elisabetta Fersini, 2020), the second shared task at the 7th evaluation campaign EVALITA 2020 (Basile et al., 2020). Given the huge amount of user-generated content on the Web, and in particular on social media, the problem of detecting, in order to possibly limit the diffusion of hate speech against women, is rapidly becoming fundamental especially for the societal impact of the phenomenon, it is very important to identify misogyny in social media.

1.1 Hate Speech (HaSpeeDe 2)

In recent years, with the acceleration of information dissemination, the identification of hate speech and offense language has become a crucial mission in multilingual sentiment analysis fields and has attracted the attention of a large number of industrial and academic researchers. From an NLP perspective, much attention has been paid to the topic of HS - together with all its possible facets and related phenomena, such as offensive/abusive language, and its identification. This is shown by the proliferation, especially in the last few years, of contributions on this topic (e.g.

[1] http://www.evalita.it/2020/tasks

Caselli et al. (2020), Jurgens et al. (2019), Fortuna et al. (2019)), corpora and lexica (e.g. de Pelle and Moreira (2017), (Sanguinetti et al., 2018), (Bassignana et al., 2018)), dedicated workshops, and shared tasks within national (GermEval [2], HASOC [3], IberLEF [4]) and international (SemEval [5]) evaluation campaigns. Among them, Gemeval2018 is about offensive language recognition and aims to promote research on offensive contents recognition in German language microblogs. The best teams system is to train three basic classifiers (maximum entropy and two random forest sets) using five disjoint feature sets and then used the maximum entropy element-level classifier for final classification (Montani and Schüller, 2018). In the SemEval-2019 shared tasks HatEval and OffensEval, HatEval is a multilingual detection of hate speech against immigrants and women on Twitter. Fermi team is the best team of Hateval. It proposes an SVM model with the RBF kernel and uses sentence embedding in Google general sentence encoder as a function (Indurthi et al., 2019). OffensEval is about the identification and classification of offensive language in social media. The NULI team is the best performing team, they use BERT-base without default parameters (Liu et al., 2019). HASOC2019 is proposed to identify hate speech and offensive content in Indo-European languages. Its purpose is to develop powerful technologies capable of processing multilingual data and to develop a transfer learning method that can utilize cross-lingual data. The optimal system is a system based on ordered neuron LSTM (ON-LSTM) and attention model and adopts the K-folding approach for ensemble (Wang et al., 2019).

1.2 Misogyny (AMI)

Unfortunately, nowadays more and more incidents of harassment against women have appeared and misogynistic comments have been found in social media, where misogynists hide behind by anonymity security. Therefore, it is very important to identify misogyny in social media. Pamungkas et al. (2020) conducted extensive and in-depth research on online misogyny, developed a state-of-the-art model for detecting misogyny in social media and explored the feasibility of detecting misogyny in a multilingual environment. Aiming at the TRAC-2 shared tasks of Aggression Identification and Misogynistic Aggression Identification, Samghabadi et al. (2020) propose an end-to-end neural model using attention on top of BERT that incorporates a multi-task learning paradigm to address both the sub-tasks simultaneously. Arango et al. (2019) discussed the implications for current research and re-conduct experiments, a closer look at model validation to give a more accurate picture of the current state-of-the-art methods. Recent investigations studied how the misogyny phenomenon takes place, such as Farrell et al. (2019) study this phenomenon by investigating the flow of extreme language across seven online communities on Reddit. Goenaga et al. (2018) automatic misogyny identification using neural networks. Automatic misogyny identification in Twitter has been firstly investigated by Anzovino et al. (2018).

2 Task and Data description

2.1 Task description

In this part, we describe one of the subtasks HaSpeeDe 2 participating in EVALITA 2020. This task introduces its novelty from three main aspects (Language variety and test of time, Stereotypical communication, Syntactic realization of HS). We participated in Task A - Hate Speech Detection (Main Task), a binary classification task aimed at determining the presence or the absence of hateful content in the text towards a given target (among Immigrants, Muslims or Roma people).

The AMI shared task proposes that misogynous content in Italian is automatic identification in Twitter. It is organized according to two main subtasks, namely subtask A - Misogyny & Aggressive Behaviour Identification and subtask B - Unbiased Misogyny Identification. We participate in subtask A, the system must recognize whether the text is misogyny, and if it is misogyny, it must also recognize whether it expresses an aggressive attitude.

2.2 Data description

HaSpeeDe 2 task organizer provides a new **HS training dataset** (binary task) based on Twitter data, accompanied by a test set including both in-domain and out-of-domain data (tweets + news headlines), as well as from different time periods. The HaSpeeDe 2020 new training set already contains the Twitter dataset of HaSpeeDe 2018. The

[2] https://projects.fzai.h-da.de/iggsa/germeval/

[3] https://hasocfire.github.io/hasoc/2020

[4] http://hitz.eus/sepln2019/

[5] http://alt.qcri.org/semeval2020/

new dataset contains a total of 6,839 tweets (label 0 means **NOT HS**, label 1 means **HS**), of which **HS** contains 2,766, **NOT HS** contains 4,703, the tweets headlines test set contains 1,263 tweets, and the news headlines test set contains 500 elements. In the experimental run, the data we recommend for this task is the result of combining the Facebook dataset (training set + test set) of HaSpeeDe 2018 with the new training set of HaSpeeDe 2020, this is to analyze the influence of out-of-domain texts in the training set. The two contain a total of 10,839 comments/tweets.

The AMI organizer provided a **raw dataset** (5,000 tweets) as the training set for participants in subtask A, the **raw dataset** is a balanced dataset of tweets manually labeled according to two levels:

- Misogynous: defines if a tweet is misogynous or not misogynous. Label 0 means **Not misogynous** tweet, label 1 means **Misogynous** tweet.
- Aggressiveness: denotes the subject of the misogynistic tweet (misogynous tweet is label 1). Label 0 means **Non-aggressive** tweet, label 1 means **Aggressive** tweet. **Not misogynous** tweet (misogynous tweet is label 0) are labeled as 0 by default.

For the test set (1,000 tweets) for subtask A provided by the AMI organizer, only the annotations on the "misogynous" and "aggressiveness" fields in the **raw dataset** will consider.

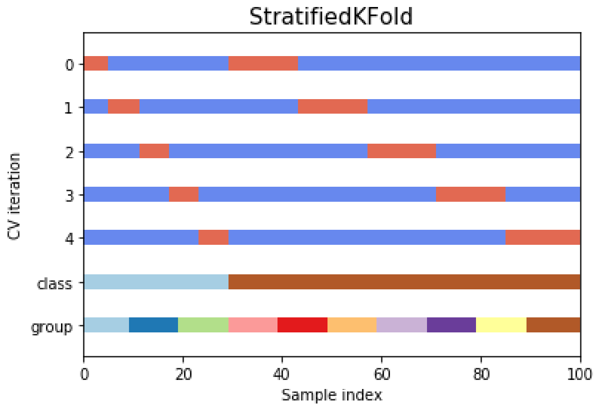

Figure 1: 5-fold stratified sampling to the training set

As shown in Figure 1, we use stratified sampling technology (StratifiedKFold), using StratifiedKFold cross-validation instead of ordinary k-fold cross-validation to evaluate a classifier. The reason is that StratifiedKFold can utilize stratified sampling to divide, which can ensure that the proportion of each category in the generated training set and validation set is consistent with the original training set so that the generated data distribution disorder will not occur. In the experiment, we used 5-fold stratified sampling. For the HaSpeeDe 2 training set (**Merged dataset**), each of which included the randomly sampled training set (8,671) and validation set (2,168). For the AMI training set (**raw dataset**), each of which included the randomly sampled training set (4,000) and validation set (1,000).

3 Description of the system

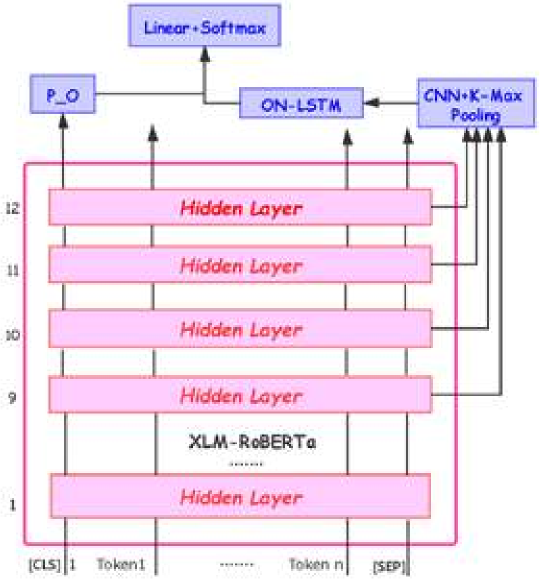

Figure 2: System architecture diagram for Task A (HaSpeeDe 2)

In this part, we introduce our final submission system. Figure 2 shows the overall framework of the system we submitted to HaSpeeDe 2 Task A. We use the pre-trained multi-language model XLM-RoBERTa. We discover the limitations of BERT's pooler output (P_O) and obtained rich semantic information by extracting the hidden state (The last four hidden layers) of XLM-RoBERTa, which is used as input for Convolution Neural Network and K-max Pooling (CNN + K-max Pooling). Then, we input the output of (CNN + K-max Pooling) into the Ordered Neurons LSTM (ON-LSTM). Finally, we concatenate the P_O and output of ON-LSTM ON-LSTM together and pass it through the Linear layer and Softmax for the final classification.

Figure 3 shows the overall framework of the system we submitted to AMI subtask A. We use the pre-trained multi-language model XLM-RoBERTa. We first get pooler output (P_O) and obtained rich semantic information by extracting the hidden state (The last four hidden layers) of XLM-RoBERTa, which is input into Ordered Neurons LSTM (ON-LSTM). Then, we input the output of ON-LSTM into Capsule Network.Finally, we concatenate the P_O and output of Capsule togetherand through the Linear layer and Softmax for the final classification.

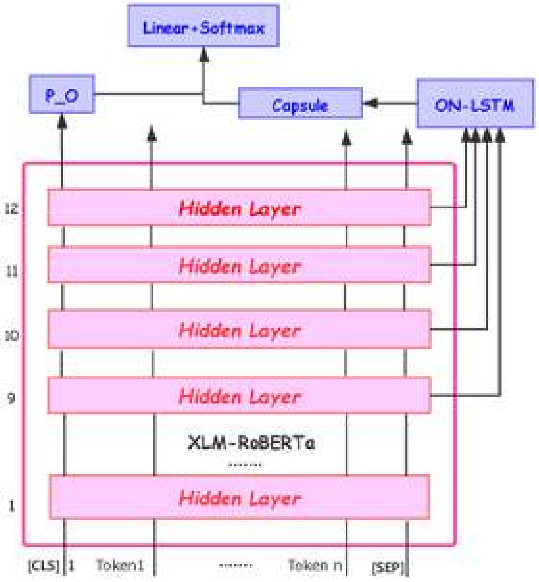

Figure 3: System architecture diagram for subtask A (AMI)

3.1 XLM-RoBERTa and hidden layer state

Early work in the field of cross-language understanding has proved the effectiveness of multilingual masked language model (MLM) in cross-language understanding, but models such as XLM (Lample and Conneau, 2019) and Multilingual BERT (Devlin et al., 2018) (pre-trained on Wikipedia) are still limited in learning useful representations of low resource languages. XLM-RoBERTa (Conneau et al., 2020) shows that the performance of cross-language transfer tasks can be significantly improved by using the large-scale multi-language pre-training model. It can be understood as a combination of XLM and RoBERTa. It is trained on 2.5 TB of newly created clean CommonCrawl data in 100 languages. Because the training of the model in this task must make full use of the whole sentence content to extract useful semantic features, which may help to deepen the understanding of the sentence and reduce the impact of noise on speech. Therefore, we use XLM-RoBERTa in this work.

In the classification task, the original output of XLM-RoBERTa is obtained through the last hidden state of the model. However, the output usually does not summarize the semantic content of the input. Recent studies have shown that abundant semantic information features are learned by the top hidden layer of BERT (Jawahar et al., 2019), which we call the semantic layer. In our opinion, the same is true of XLM-RoBERTa. Therefore, in order to make the model obtain more abundant semantic information features, we propose the system as shown in Figure 2 for HaSpeeDe 2 Task A. Firstly, we get P_O. Secondly, we extract the hidden state of the last four layers of XLM-RoBERTa and input them into CNN and K-max Pooling. Then, input into ON-LSTM. For AMI subtask A, we propose the system as shown in Figure 3. Firstly, we get P_O. Secondly, we extract the hidden state of the last four layers of XLM-RoBERTa and input them into ON-LSTM. Then, input into Capsule.

3.2 CNN and K-max Pooling

As shown in Figure 2, we input the extracted hidden states of the last four layers of XLM-RoBERTa into CNN and K-max Pooling for convolution operations to obtain multiple feature maps. The specific operation: a sentence contains L words, each of which has a dimension of d after the embedding layer, and the representation of the sentence is formed by splicing the L words to form a matrix of $L * d$. There are several convolution kernels in the convolutional layer, the size of which is $N * d$, and N is the filter window size. The convolution operation is to apply a convolution kernel to create a new feature in a matrix that is spliced by words. Its formula is as follows:

$$C_l = f(w * x(l : L + N - 1) + b) \qquad (1)$$

where l represents the lth word, C_l is the feature, w is the convolution kernel, b is the bias term, and f is a nonlinear function. After the convolution operation of the whole sentence, a feature map is obtained, which is a vector of size $L + N$ - 1.

Another important idea of CNN is pooling. The pooling layer is usually connected behind the convolution layer. The purpose of introducing it is to simplify the output of the convolutional layer and perform dimensionality reduction on the features of the Filter to form the final feature. Here is the K-max Pooling operation, which takes the value of the scores in Top K among all the feature values, and retains the original order of these feature values, that is, by retaining some feature information for subsequent use. Obviously, K-max Pooling can express the same type of feature multiple times, that is, can express the intensity of a certain type of feature; in addition, because the relative order of these Top K eigenvalues is preserved, it should be said that it retains part of the position information. However, this location information is only the relative order between features, not absolute location information.

3.3 Ordered Neurons LSTM

For HaSpeeDe 2, as shown in Figure 2, we input the output of CNN and K-max pooling into ON-LSTM. For AMI, as shown in Figure 3, We input the extracted hidden states of the last four layers of XLM-RoBERTa into ON-LSTM. ON-LSTM is a new variant of LSTM, which sorts the neurons in a specific order, allowing the hierarchical structure (tree structure) to be integrated into the LSTM to express richer information. The gate structure and output structure of ON-LSTM are still similar to the original LSTM. The difference is that the update mechanism from $\widehat{c_t}$ to c_t is different. The formula is as follows (Shen et al., 2018):

$$\widetilde{f}_t = \overrightarrow{cs}(softmax(W_{\widetilde{f}}x_t + U_{\widetilde{f}}h_{t-1} + b_{\widetilde{f}}) \quad (2)$$

$$\widetilde{i}_t = \overleftarrow{cs}(softmax(W_{\widetilde{i}}x_t + U_{\widetilde{i}}h_{t-1} + b_{\widetilde{i}}) \quad (3)$$

$$w_t = \widetilde{f}_t \circ \widetilde{i}_t \quad (4)$$

$$\begin{aligned} c_t =& w_t \circ (f_t \circ c_{t-1} + i_t \circ \widehat{c_t}) + (\widetilde{f}_t - w_t) \\ & \circ c_{t-1} + (\widetilde{i}_t - w_t) \circ \widehat{c_t} \end{aligned} \quad (5)$$

Among them, $\overrightarrow{cs}$ and $\overleftarrow{cs}$ are cumsum() operations in the right and left directions, respectively. the newly introduced $\widetilde{f}_t$ and $\widetilde{i}_t$ represent the master forget gate and master input gate respectively. w_t represents a vector where the intersection part is 1 and the rest is all 0. In this way, the high-level information remains a considerable long distance, while the low-level information may be updated at each step of input, thereby embedding the hierarchical structure through information grading.

3.4 Capsule Network

As shown in Figure 3, we input the output of ON-LSTM into Capsule. In the deep learning model, spatial patterns are aggregated at a lower level, which helps to represent higher-level concepts. We use the Capsule Network (Sabour et al., 2017) to enhance the models feature extraction capabilities, spatial insensitivity methods are inevitably limited by the abundant text structure (such as saving the location of words, semantic information, grammatical structure, etc.), difficult to effectively encode, and lack of text expression ability. The Capsule network effectively improved this disadvantage by using neuron vectors instead of individual neuron nodes of traditional neural networks to train this new neural network in the dynamic routing way. The Capsule's parameter update algorithm is routing-by-agreement, a lower-level capsule prefers to send its output to higher-level capsule whose activity vectors have a big scalar product with the prediction coming from the lower-level capsule. The calculation formula of the Capsule is as follows:

$$V_j = \frac{\parallel S_j \parallel^2}{1+ \parallel S_j \parallel^2} \frac{S_j}{\parallel S_j \parallel} \quad (6)$$

$$S_j = \sum_i C_{ij}\hat{u}_{j|i}, \qquad \hat{u}_{j|i} = W_{ij}u_i \quad (7)$$

where V_j is the vector output of capsule j and S_j is its total input, prediction vectors $\hat{u}_{j|i}$ is by multiplying the output u_i of a capsule in the layer below by a weight matrix W_{ij}, the C_{ij} are coupling coefficients that are determined by the iterative dynamic routing process.

The most fundamental difference between the Capsule network and the traditional artificial neural network lies in the unit structure of the network. For traditional neural networks, the calculation of neurons can be divided into the following three steps: 1. Perform a scalar weighted calculation on the input. 2. Sum the weighted input scalars. 3. Nonlinearization from scalar to the scalar. For the Capsule, its calculation is divided into the following four steps: 1. Do matrix multiplication on the input vector. 2. Scalar weighting of the input vector. 3. Sum the weighted vector. 4. Vector-to-vector nonlinearization. The biggest difference between the Capsule network and the traditional neural network is the unit output. The output of the traditional neural network is a value, while the output of the Capsule network is a vector, which can contain abundant features and is more interpretable.

3.5 Experiment setting

For the XLM-RoBERTa, we use XLM-RoBERTa-base[6] pre-trained model, which contains 12 layers. We use Binary cross-entropy, Adam optimizer with a learning rate of 5e-5. The batch size is set to 32 and the max sequence length is set to 80. We extract the hidden layer state of XLM-RoBERTa by setting the output_hidden_States is true. The model is trained in 8 epochs with a dropout rate of 0.1.

For the Convolution Neural Network,we use 2D convolution (nn.Conv2d[7]). The size of the convolution kernel is set to (3,4,5) and the number of convolution kernels is set to 256.

For the ON-LSTM, we set the hidden units to 128 and num levels to 16.

For the Capsule Network, we set num capsule to 10, dim capsule to 16, routings to 4.

4 Results and Discussion

Task	Our Score	Best Score	Rank
HaSpeeDe	Macro F1		
Tweets	0.7717	0.8088	8
News	0.6922	0.7744	7
AMI	Average F1		
subtask A	0.7313	0.7406	3

Table 1: Classification results of our best runs on the HaSpeeDe 2 Task A and AMI subtask A.

Table 1 reports the official results of the best runs on the two tasks we participate in. For these two tasks, we submitted the results of two runs, and the results of both runs were ideal and equally matched. In the following subsections, the results obtained in each task will be discussed.

XLM-RoBERTa with only P_O in News The validation set of 1-fold				
Category	**P**	**R**	**F1**	**Instances**
Not Hate	0.70	0.981	0.817	1355
Hate	0.886	0.259	0.401	813
Macro F1	0.793	0.62	0.609	2168
XLM-RoBERTa with only P_O in Tweets The validation set of 1-fold				
Category	**P**	**R**	**F1**	**Instances**
Not Hate	0.805	0.569	0.667	1355
Hate	0.659	0.858	0.745	813
Macro F1	0.723	0.713	0.706	2168

Table 2: Precision, Recall, F1 score and Instances for XLM-RoBERTa with only P_O in HaSpeeDe 2 Task A (The validation set is the first fold in the 5-fold stratified cross-validation)

The number of different hidden layers of XLM-RoBERTa (The validation set of 1-fold)		
Systems Hidden layers	**HS-News** Macro F1	**HS-Tweets** Macro F1
The last layers	0.623	0.725
The last two layers	0.646	0.734
The last three layers	0.66	0.749
The last four layers	0.703	0.798

Table 3: The performance of our model at different hidden layers (The validation set is the first fold in the 5-fold stratified cross-validation)

4.1 HaSpeeDe 2 Task A

In our experiment, we find the limitations of P_O for sentiment analysis of hate text in Italian languages. In the classification task, the original output of BERT is P_O. In the same way, we just put P_O as the output of XLM-RoBERTa.The results are shown in Table 2. We can see that the results are not good when only P_O is used as the output of XLM-RoBERTa. We think that just using P_O as the output will lose some effective semantic information. So we think that deep and abundant semantic features are effective for this work. We extract the hidden state of XLM-RoBERTa and we also discover that the performance of the model improves with the increase of the semantic layer. Table 3 shows the performance of our model at different semantic layers. Table 4 shows our results on the test set.

4.2 AMI subtask A

In this work, we have similar tasks as discussed in Section 4.1, and we consider the influence of P_O for identifying misogyny content. We conduct experiments on the AMI subtask A base on the mod-

[6] https://huggingface.co/xlm-roberta-base

[7] https://pytorch.org/docs/stable/generated/torch.nn.Conv2d

The last four hidden states of XLM-RoBERTa				
News	**P**	**R**	**F1**	**Macro F1**
Not Hate Hate	0.7486 0.7203	0.8965 0.4696	0.8159 0.5685	**0.6922**
Tweets	**P**	**R**	**F1**	**Macro F1**
Not Hate Hate	0.8037 0.7448	0.7285 0.8167	0.7643 0.7791	**0.7717**

Table 4: Results of Macro F1 on Test set

el in HaSpeeDe 2, and in order to improve the performance, we propose a new method base on this model. Table 5 shows the comparative experimental data of the CNN + K-max Pooling + ON-LSTM method and the ON-LSTM + Capsule method. Table 6 shows the results of our new model for AMI subtask A on the test set. Run 1 only extracts the last four hidden layer states of XLM-RoBERTa and inputs them into ON-LSTM, then through the Capsule Network, and finally performs classification (without using P_O). Run 2 is to concatenate the output of the Capsule Network with the obtained P_O and input it to the classifier for final classification (using P_O). We think that concatenate the P_O and the hidden layer will retain richer semantic information and show excellent results.

Base on XLM-RoBERTa model (The validation set of 1-fold)	
Method	**Macro F1**
CNN + K-max Pooling + ON-LSTM (HaSpeeDe 2 Model)	0.786
ON-LSTM + Capsule (AMI model)	0.857

Table 5: Comparison of experimental data between CNN + K-max Pooling method and ON-LSTM + Capsule method on the validation set. (The validation set is the first fold in the 5-fold stratified cross-validation)

5 Conclusion

In the experiment, we find the limitation of only using pooler output as the XLM-RoBERTa's output. To obtain deeper and more abundant semantic features, we extract the hidden layer state of XLM-RoBERTa. The result shows that it is helpful to improve the performance of XLM-RoBERTa to obtain more abundant semantic information features by extracting the hidden state of XLM-RoBERTa. We test the effects of using the external dataset (**Merged dataset**) and not using the external dataset (**raw dataset**). Our conclusion is that using data from the same social network for training and test is a necessary condition for good performance. In addition, adding data from different social networks can improve results.

System	Average F1
Run 1 (without using P_O)	0.7014
Run 2 (using P_O)	0.7313

Table 6: The results on the test set for AMI subtask A

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.

Aymé Arango, Jorge Pérez, and Barbara Poblete. 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 45–54.

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.

Elisa Bassignana, Valerio Basile, and Viviana Patti. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6. CEUR-WS.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the evalita 2018 hate speech detection task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, pages 1–9. CEUR.

Tomasso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso. 2018. Sixth evaluation campaign of natural language processing and speech tools for italian: Final workshop (evalita 2018). In *EVALITA 2018*. CEUR Workshop Proceedings (CEUR-WS. org).

Tommaso Caselli, Valerio Basile, Jelena Mitrović, Inga Kartoziya, and Michael Granitzer. 2020. I feel offended, dont be abusive! implicit/explicit messages in offensive and abusive language. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6193–6202.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Rogers Prates de Pelle and Viviane P Moreira. 2017. Offensive comments in the brazilian web: a dataset and baseline results. In *Anais do VI Brazilian Workshop on Social Network Analysis and Mining*. SBC.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Paolo Rosso Elisabetta Fersini, Debora Nozza. 2020. Overview of the evalita 2020 automatic misogyny identification (ami) task. 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.

Tracie Farrell, Miriam Fernandez, Jakub Novotny, and Harith Alani. 2019. Exploring misogyny across the manosphere in reddit. In *Proceedings of the 10th ACM Conference on Web Science*, pages 87–96.

Paula Fortuna, Joao Rocha da Silva, Leo Wanner, Sérgio Nunes, et al. 2019. A hierarchically-labeled portuguese hate speech dataset. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 94–104.

Iakes Goenaga, Aitziber Atutxa, Koldo Gojenola, Arantza Casillas, Arantza Díaz de Ilarraza, Nerea Ezeiza, Maite Oronoz, Alicia Pérez, and Olatz Perez-de Viñaspre. 2018. Automatic misogyny identification using neural networks. In *IberEval@ SEPLN*, pages 249–254.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Nikhil Chakravartula, Manish Gupta, and Vasudeva Varma. 2019. Fermi at semeval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 70–74.

Ganesh Jawahar, Benot Sagot, and Djam Seddah. 2019. What does bert learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

David Jurgens, Eshwar Chandrasekharan, and Libby Hemphill. 2019. A just and comprehensive strategy for using nlp to address online abuse. *arXiv preprint arXiv:1906.01738*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining.

Ping Liu, Wen Li, and Liang Zou. 2019. Nuli at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91.

Joaquın Padilla Montani and Peter Schüller. 2018. Tuwienkbs at germeval 2018: German abusive tweet detection. In *14th Conference on Natural Language Processing KONVENS*, volume 2018, page 45.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2020. Misogyny detection in twitter: a multilingual and cross-domain study. *Information Processing & Management*, 57(6):102360.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules.

Niloofar Safi Samghabadi, Parth Patwa, PYKL Srinivas, Prerana Mukherjee, Amitava Das, and Thamar Solorio. 2020. Aggression and misogyny detection using bert: A multi-task approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 126–131.

Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An italian twitter corpus of hate speech against immigrants. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Manuela Sanguinetti, Gloria Comandini, Elisa Di Nuovo, Simona Frenda, Marco Stranisci, Cristina Bosco, Tommaso Caselli, Viviana Patti, and Irene Russo. 2020. HaSpeeDe 2@EVALITA2020: Overview of the EVALITA 2020 Hate Speech Detection Task. 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.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks.

Bin Wang, Yunxia Ding, Shengyan Liu, and Xiaobing Zhou. 2019. Ynu_wb at hasoc 2019: Ordered neurons lstm with attention for identifying hate speech and offensive language. In *FIRE (Working Notes)*, pages 191–198.