2018 12th IEEE International Conference on Semantic Computing

Movie Genre Classification from Plot Summaries using Bidirectional LSTM

Ali Mert Ertugrul
Graduate School of Informatics
Middle East Technical University
Ankara, Turkey
alimert@metu.edu.tr

Pinar Karagoz
Computer Engineering Department
Middle East Technical University
Ankara, Turkey
karagoz@ceng.metu.edu.tr

***Abstract*—Movie plot summaries are expected to reflect the genre of movies since many spectators read the plot summaries before deciding to watch a movie. In this study, we perform movie genre classification from plot summaries of movies using bidirectional LSTM (Bi-LSTM). We first divide each plot summary of a movie into sentences and assign the genre of corresponding movie to each sentence. Next, using the word representations of sentences, we train Bi-LSTM networks. We estimate the genres for each sentence separately. Since plot summaries generally contain multiple sentences, we use majority voting for the final decision by considering the posterior probabilities of genres assigned to sentences. Our results reflect that, training Bi-LSTM network after dividing the plot summaries into their sentences and fusing the predictions for individual sentences outperform training the network with the whole plot summaries with the limited amount of data. Moreover, employing Bi-LSTM performs better compared to basic Recurrent Neural Networks (RNNs) and Logistic Regression (LR) as a baseline.**

***Index Terms*—Movie genre classification; LSTM; Recurrent Neural Networks (RNNs)**

I. Introduction and Background

Movie plot summaries reflect the genre of the movies such as action, drama, horror, etc., such that people can easily capture the genre information of the movies from their plot summaries. Especially, several sentences in the plot summaries are high representatives of genre of the movie. People usually read the plot summaries of movies before watching them to get an idea about the movie. Therefore, plot summaries are written in such a way that they convey the genre information to the people. For example, if the plot mentions humorous obstacles that must be overcome before lovers eventually come together, the movie is a likely to be a *romantic-comedy* [1]. In this regard, there is a hidden representation of genre information in the movie plot summaries. In this study, we aim to learn this hidden representation. In other words, our purpose is to classify the genres of the movies from their plot summaries using Bi-LSTM by considering genre information represented by each individual sentence. With this method, representations of plot summaries can be used for movie recommendation. In addition to that, it can be inferred whether a plot summary actually reflects the genre of the movie it belongs to. Therefore, this method can be beneficial during the preparation of movie plots.

In the literature, there exists a number of studies that perform movie genre classification using a variety of sources including visual, audio and textual features from trailers, posters and texts. Among the studies that employ visual and/or audio features, Rasheed et al. [2], [3] utilized visual features including average shot length, color variance, motion content and lighting key, from movie previews to predict movie genres. Yuan et al. [4] also employed visual features from videos including temporal and spatial ones to classify genres using hierarchical SVM. Zhou et al. [5] represented movie trailers using bag-of-visual-words model with shot classes as vocabularies and utilized them for genre classification. Moreover, Huang et al. [6] extracted both visual and audio features from movie trailers using a meta-heuristic optimization algorithm and performed genre classification. Ekenel et al. [7] combined low level audio and visual features including signal energy, fundamental frequency for audio; color and texture-based features for visual representation to conduct multi-modal genre classification. Ivasic et al. [8] employed low-level visual features based on colors and edges obtained from movie posters, then used them to classify posters into genres. Furthermore, Simoes et al. [9] and Wehrmann et al. [10] used convolutional neural networks (CNNs) based architectures to perform movie genre classification from movie trailers instead of using hand-crafted features.

In addition to efforts employing visual and audio features, several studies used textual sources including plots and synopses for movie genre classification. Fu et al. [11] utilized vector space model to represent synopses and used this representation as input for SVM. Hong et al. [12] extracted textual features from social tags via social websites. Then, they applied probabilistic latent semantic analysis (PLSA) to incorporate textual, visual and audio features for genre classification. Furthermore, Arevalo et al. [13] proposed a gated unit for multi-modal classification task and they performed movie genre classification using poster and plot information with a basic recurrent neural networks (RNNs) to represent plot information. Similarly, Pham et al. [14] proposed column network for collective classification and they evaluated this network on movie genre classification task using plot summaries. They represented plot summaries as Bag-of-Words (BoW) vector of 1.000 most frequent words. Aforementioned studies using

0-7695-6360-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICSC.2018.00043

TABLE I
DISTRIBUTION OF THE SAMPLES FOR EACH GENRE

	Thriller	Horror	Comedy	Drama
# of Full Plots	1590	1590	1590	1590
# of Sentences	5421	5437	5480	5940

plots or synopses either did not benefit from the power of deep learning for sequence modeling of textual data or they obtained textual representations using basic RNNs. Moreover, these studies performed document-level use of text for genre classification.

II. METHOD

A. Data Collection and Pre-processing

In preprocessing step, we first obtained movie names from MovieLens[1] datasets. We further collected necessary information of the movies including full plot summaries (input) and genres (ground-truth) through OMDb API[2] using corresponding movie names as inputs.

Within the scope of this study, we selected four types of genres, namely *Thriller*, *Horror*, *Comedy* and *Drama* for movie genre classification. Since the number of the movies in the database vary for each genre, we randomly sampled movies for each of them uniformly. However, the total number of sentences in the plot summaries changes for each genre as the plots may include different number of sentences. Accordingly, in the document-level classification task (using whole plots as inputs), we have uniformly sampled the data based on their genres. On the other hand, the data for the sentence-level classification (using sentences as inputs), is unbalanced for the training. As a result, we obtained a total of 6.360 movies and 22.278 sentences for the genre classification task, respectively. The Table I shows the distribution of the number of the movies and the total number of sentences for each genre in the dataset.

Before training a model for classification, we conducted a pre-processing step. We first converted all texts in the plots to lowercase. Next, we eliminated all punctuation marks except the ones that separate the sentences. Additionally, we eliminated the stop-words. We also divided plot summaries into sentences for the sentence-level classification task. We performed all tasks in pre-processing step using NLTK[3].

B. Text Representation

The purpose of this step is to represent semantic and syntactic relationship among the words, which improves the performance where the training data is limited. After pre-processing step, each input (full plot for document-level and sentence for sentence-level) is represented using continuous vector representation. In order to do that, the pre-trained word vectors that are proposed by [15] are used. The word vectors are obtained as a result of training on Wikipedia. These vectors

[1]https://grouplens.org/datasets/movielens/
[2]http://www.omdbapi.com/
[3]http://www.nltk.org/

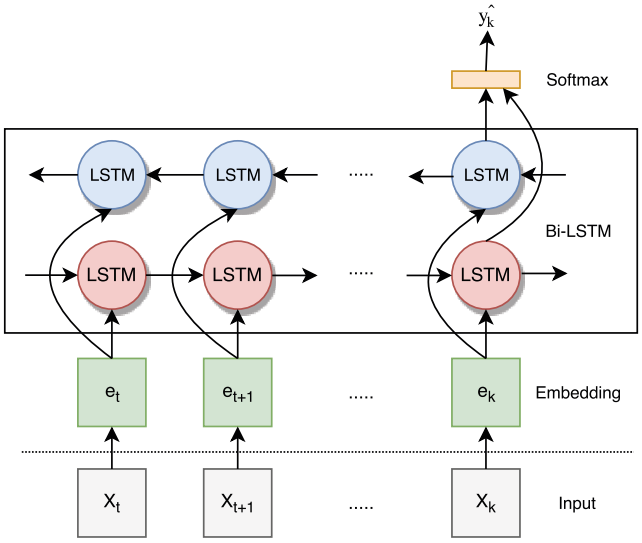

Fig. 1. Bi-LSTM Network Architecture for Movie Genre Classification

are in dimension of 300 and they were obtained using the skip-gram model. Therefore, the relationships between words and their context are modeled beforehand. As a result, the row input is converted to continuous vector representation and then fed into the network. Note that, for any word in the plots that does not have a corresponding word vector in the dictionary, a random word vector in dimension of 300 was generated.

C. Model

The LSTM model [16] is an RNN architecture, which is capable of learning complex dependencies across time. LSTM RNNs address the vanishing gradient problem of basic RNNs by employing gating functions together with the state dynamics. In this study, we use Bi-LSTM network. It is composed of two LSTM neural networks, a forward LSTM to model the preceding contexts, and a backward LSTM to model the following contexts respectively. The architecture used in the study is given in Fig. 1.

Note that, each plot summary of a movie is divided into sentences and the genre of corresponding movie is assigned to each sentence. During training, each input (sentence) is represented as the words it includes and continuous word representations are obtained using [15]. It is useful when the limited data is used since semantic and syntactic relationship among the words are captured. We name this representation in the architecture in Fig. 1 as embedding layer. Then, the word representations are fed into the Bi-LSTM network. Practically, a linear projection layer is put between Bi-LSTM and softmax layers. Finally, a softmax layer, which is stacked on the top of the Bi-LSTM, takes the learned representations of the last output of Bi-LSTM as the input, and returns the classification probabilities for each movie genre.

In order to train the model, we minimize the negative log-likelihood of the estimation error, where the loss function is given in Eq. 1 below.

$$L(\theta) = -\frac{1}{C}\sum_{i=1}^{C} y_i \log(\hat{y}_i), \quad (1)$$

where C is the number of the target classes, y is the one-hot representation of the ground truth, and $\hat{y}$ is the estimated probability distribution assigned to the genres by the model.

D. Classification

Since we divide the plot summaries into sentences, we estimate the class labels for each of them separately during test time. However, we need to assign a single class label for any given plot summary. Therefore, we fuse the decisions of the model for each sentence to obtain a final class label for the corresponding plot summary. For that purpose, we use majority voting to obtain the final decision by considering the posterior probabilities of genres assigned to the sentences. If the majority voting outputs a single label, then the plot summary is assigned to that label. On the other hand, if more than one genre have the maximum number of votes, we assign the genre to the plot summary, whose average class posterior probability is the maximum among others. The steps of the genre label assignment process are given in Algorithm 1.

Algorithm 1 Assignment of genre label to a plot summary during prediction.

Require: $\mathrm{G}_p \in \mathbb{R}^{n \times c}$, genre posterior probability vectors of all sentences of plot summary p, where n is the number of sentences of p and c is the number of target classes.

Require: $\mathrm{G}_{p,s} \in \mathbb{R}^{c}$, genre posterior probability vector of sentence s of plot summary p.

Ensure: $\hat{y}_p$, estimated genre label for plot summary p.

```
G_p^avg ← getAverageProbabilities(G_p)
label_count ← zeros(c)
for s ∈ p do
    estimated ← index(max(G_{p,s}))
    label_count(estimated) ← label_count(estimated) + 1
end for
candidate_labels ← indices(max(label_count))
if candidate_labels.length() > 1 then
    ŷ_p ← findLabelOfHighestProb(G_p^avg, candidate_labels)
else
    ŷ_p ← candidate_labels.first()
end if
```

E. Evaluation Measures

Since the movie genre classification task is a multi-class classification problem, we used two versions for the averages of the *f-score* (f_1), which are *micro* and *macro*. The former computes the *f-score* using all estimations at once. On the other hand, the latter computes the *f-score* for each genre separately and then averages the results. The calculations of *micro* precision and *micro* recall are given in Eq. 2a and 2b whereas equations for *macro* precision and *macro* recall are shown in Eq. 3a and 3b.

$$p^{micro} = \frac{\sum_{i=1}^{C} tp_i}{\sum_{i=1}^{C} tp_i + \sum_{i=1}^{C} fp_i} \quad (2a)$$

$$r^{micro} = \frac{\sum_{i=1}^{C} tp_i}{\sum_{i=1}^{C} tp_i + \sum_{i=1}^{C} fn_i} \quad (2b)$$

$$p^{macro} = \frac{1}{C}\sum_{i=1}^{C} \frac{tp_i}{tp_i + fp_i} \quad (3a)$$

$$r^{macro} = \frac{1}{C}\sum_{i=1}^{C} \frac{tp_i}{tp_i + fn_i} \quad (3b)$$

where C is the number of target labels, p is the precision, r is the recall, tp_i , fp_i and fn_i stand for the number of true positives, false positives and false negatives for the i^{th} target label, respectively.

For both *micro* and *macro* measures, we compute the *f-score* as $f_1 = \frac{2 \times p \times r}{p+r}$. Note that, since we perform our experiments on a single dataset, *micro* precision, *micro* recall and *micro f-score* values are all equal and they represent the accuracy of the classifier. Accordingly, we only present the *micro f-score* for the *micro* results.

III. Experiments and Results

In this study, we perform multi-class movie genre classification from plot summaries using Bi-LSTM where the class labels are *Thriller*, *Horror*, *Comedy* and *Drama*. We predict the genre of a movie by combining the decisions given for sentences of its plot summary, which is called *sentence-level* approach. On the other hand, when we use the whole plot summary for training without dividing it into its sentences, we call it as *document-level* approach.

In order to measure the performance of the proposed method, we compare it with two baseline methods under different settings. First, we train an ordinary RNNs model using sentence-level and document-level approaches. We use the same representations used for our method while training RNNs model. Second, we train a logistic regression (LR) classifier using both settings in a similar way. While training LR model, we obtain Bag-of-Words (BoW) representation for each plot summary or sentence for *document-level* or *sentence-level* approach, respectively. Then we fill the vector of plot summary or sentence with term frequency inverse document frequency (TD-IDF) [17] values. IDF weight for each word is calculated using only training dataset. There exist 27.336 unique words in the training dataset. Accordingly, each input is represented by 27.336 dimensional feature vector. Moreover, we also compare the results of our method with Bi-LSTM model trained with *document-level* approach. Note that, for all methods, the same pre-processing step is applied explained in Section II-A.

TABLE II
MOVIE GENRE CLASSIFICATION RESULTS (%) MEASURED BY PRECISION, RECALL AND F-SCORE

	Macro Pre.	**Macro Rec.**	**Macro f_1**	**Micro f_1**
Bi-LSTM-s	**67.75**	**67.61**	**67.68**	**67.61**
Bi-LSTM-d	65.92	64.62	65.26	64.62
RNN-s	62.79	62.42	62.61	62.42
RNN-d	57.72	55.50	56.59	55.50
LR-s	63.05	62.74	62.89	62.74
LR-d	64.03	63.84	63.94	63.84

Setup. We divide the dataset into training (70%), validation (15%) and test (15%) subsets. For each method, we select the model that performs the best on the validation set and report its performance on the test set. We train all Bi-LSTM networks and RNNs with stochastic gradient descent (SGD) using Adam optimizer [18]. The learning rate is set to 0.005. For both Bi-LSTM and RNNs experiments, the hidden layer size and number of hidden units are set to $\{1, 2, 3\}$ and $\{8, 16, 32, 64, 128\}$, respectively. Moreover, the regularization parameter C for LR experiment is set to $\{0.01, 0.1, 1, 10, 100\}$.

Table II shows the results of the experiments. Where suffix s represents the *sentence-level* approach, suffix d stands for *document-level* approach. According to the results, our method, Bi-LSTM-s, significantly outperforms the other methods in terms of both *macro f-score* and *micro f-score* which are 67.78% and 67.61%, respectively. We also observe that *sentence-level* approach importantly boosts the performance when the recurrent neural networks are used for the classification. Bi-LSTM-s and RNN-s perform superior than the *document-level* settings of the same networks. On the other hand, *document-level* approach gives slightly better performance compared to *sentence-level* approach when LR is used for training. The reason for this can be the way of representation we use for the inputs. Since we represent the inputs using BoW model, the representations are sparser in *sentence-level* approach. This may prevent the classifier to learn satisfactorily in case of limited data for training.

We also share the values of *precision*, *recall* and *f-score* of Bi-LSTM-s method for each genre in Table III. According to the results, the proposed method performs better while estimating the genre of *Horror*. On the other hand, the lowest performance is obtained while predicting the genre of *Thriller*.

TABLE III
PRECISION, RECALL AND F-SCORE VALUES (%) FOR EACH GENRE OBTAINED USING BI-LSTM-S

	Precision	**Recall**	**F-score**
Thriller	68.46	55.98	61.59
Horror	76.22	78.62	77.4
Comedy	64.71	69.18	66.87
Drama	61.63	66.67	64.05

IV. CONCLUSION

In this study, we perform movie genre classification from plot summaries using Bi-LSTM network. Instead of using whole plot summary as input, we divide it into its sentences and train the network using those sentences. During prediction, we fuse the decisions of the model for each sentence to obtain a final class label. Results show that our method significantly outperforms ordinary RNNs and LR. Also, we observe that using sentences to label the genre of a movie performs better than using whole plot summary for recurrent neural networks when the data is limited. As a future work, we are planning to increase the size of the dataset and extend our method for multi-label, multi-class movie genre classification.

REFERENCES

[1] T. B. Cargal, *Hearing a film, seeing a sermon: Preaching and popular movies*. Westminster John Knox Press, 2007.

[2] Z. Rasheed, Y. Sheikh, and M. Shah, "Semantic film preview classification using low-level computable features," in *3rd International Workshop on Multimedia Data and Document Engineering (MDDE-2003)*, 2003.

[3] ——, "On the use of computable features for film classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 52–64, 2005.

[4] X. Yuan, W. Lai, T. Mei, X.-S. Hua, X.-Q. Wu, and S. Li, "Automatic video genre categorization using hierarchical svm," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 2905–2908.

[5] H. Zhou, T. Hermans, A. V. Karandikar, and J. M. Rehg, "Movie genre classification via scene categorization," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 747–750. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874068

[6] Y.-F. Huang and S.-H. Wang, "Movie genre classification using svm with audio and video features," *Active Media Technology*, pp. 1–10, 2012.

[7] H. K. Ekenel and T. Semela, "Multimodal genre classification of tv programs and youtube videos," *Multimedia tools and applications*, vol. 63, no. 2, pp. 547–567, 2013.

[8] M. Ivašić-Kos, M. Pobar, and L. Mikec, "Movie posters classification into genres based on low-level features," in *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014*, 2014.

[9] G. S. Simões, J. Wehrmann, R. C. Barros, and D. D. Ruiz, "Movie genre classification with convolutional neural networks," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 259–266.

[10] J. Wehrmann and R. C. Barros, "Convolutions through time for multi-label movie genre classification," in *Proceedings of the Symposium on Applied Computing*. ACM, 2017, pp. 114–119.

[11] Z. Fu, B. Li, J. Li, and S. Wei, "Fast film genres classification combining poster and synopsis," in *International Conference on Intelligent Science and Big Data Engineering*. Springer, 2015, pp. 72–81.

[12] H.-Z. Hong and J.-I. G. Hwang, "Multimodal plsa for movie genre classification," in *International Workshop on Multiple Classifier Systems*. Springer, 2015, pp. 159–167.

[13] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017.

[14] T. Pham, T. Tran, D. Q. Phung, and S. Venkatesh, "Column networks for collective classification." in *AAAI*, 2017, pp. 2485–2491.

[15] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of research and development*, vol. 1, no. 4, pp. 309–317, 1957.

[18] D. P. Kingma and J. B. Adam, "A method for stochastic optimization. 2014," *arXiv preprint arXiv:1412.6980*.