The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Self-Attention for Synopsis-Based Multi-Label Movie Genre Classification

Jônatas Wehrmann, Maurício A. Lopes, Rodrigo C. Barros
Machine Intelligence and Robotics Research Group
Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681, 90619-900, Porto Alegre, RS, Brazil
{jonatas.wehrmann@acad., mauricio.armani@acad., rodrigo.barros@}pucrs.br

Abstract

This paper proposes a novel neural network architecture for multi-label movie genre classification based on the textual synopsis of the movie. We design an architecture that transforms the synopsis into a $n \times d$ matrix, in which n is the temporal dimension (total number of words in the synopsis, indicating the directional flow of the words) and d is the word-embedding vector that densely projects the respective word onto a high-dimensional feature space. A self-attention mechanism is employed to automatically learn the importance of the features in each temporal step, so the complex mapping from synopsis to a given genre (or set of genres) can be properly performed. Experiments show that our approach outperforms state-of-the-art methods for text classification based on neural networks in the largest movie genre dataset (LMTD).

1 Introduction

Neural networks are nowadays known to be the state-of-the-art method for many image, video, audio, and text based tasks, such as supervised image classification, localization, detection, semantic segmentation, speech recognition, text classification, text summarization, translation, just to name a few. Borrowing concepts from neuroscience, artificial neural networks comprise a mathematical schema capable of assigning meaning to what is seen, heard, or read, being known as an effective method for performing *representation learning* over unstructured data.

Neural networks consist of multiple layers of sets of neurons that process (portions of the) input data, hierarchically learning concepts in sequence to allow complex mappings from input to desired output (Goodfellow, Bengio, and Courville 2016). The widely-known backpropagation algorithm is often the desired choice for training those networks by following the chain rule of derivatives of a loss function with respect to the network parameters.

In this paper, we investigate the use of neural networks for automatically classifying movies according to their genre (e.g., action, horror, drama, comedy). Unlike previous work on the subject (Simões et al. 2016; Wehrmann and Barros 2017b; Wehrmann et al. 2016), which make use of images, video, and audio extracted from the movies or from the movie trailers, we address the problem of assigning one or multiple genres for a movie based on its textual synopsis. Hence, we are dealing with an instance of the traditional multi-label text classification problem, in which the goal is to map a textual input (movie synopsis) into one or multiple classes (genres of movies).

For properly addressing the problem, we propose a novel neural network architecture, which considers each input as a $n \times d$ matrix, in which n is the temporal dimension (total number of words in the synopsis, indicating the ordered flow of the words) and d is the word-embedding vector that densely projects the respective word onto a high-dimensional feature space. We employ a self-attention mechanism so the network can automatically learn the importance of the features from each word in each temporal step, so the complex mapping from synopsis to a given genre (or set of genres) can be effectively performed.

We compare our novel approach with several state-of-the-art methods for text classification based on neural networks, such as LSTMs (Hochreiter and Schmidhuber 1997), GRUs (Chung et al. 2015), Textual Convolutions (Kim 2014), and the recent Fast Text (Joulin et al. 2016). Several experiments show that our approaches outperform all baselines in the largest and most well-known multi-label movie genre classification dataset, Labelled Movie Trailer Dataset (LMTD) (Simões et al. 2016; Wehrmann et al. 2016; Wehrmann and Barros 2017b).

This paper is organized as follows. Section 2 describes in detail our proposed approach, whereas Sections 3 and 4 present the experimental analysis that was conducted for validating our novel method. Section 5 discusses related work in the area of movie genre classification. Finally, we end this paper with our conclusions and suggestions for future work in Section 6.

2 SAS-MC

In this paper we introduce SAS-MC, a novel method for multi-label classification of movie genres based on short synopses. SAS-MC makes use of the self-attention mechanism introduced in (Lin et al. 2017), which is designed to automatically learn the importance of the features in each time-step by analyzing different views (or *hops*). Our architecture is a simple yet effective approach for learning multi-label movie genre information from textual synopses.

Formally, SAS-MC is designed to learn a function ϕ, so

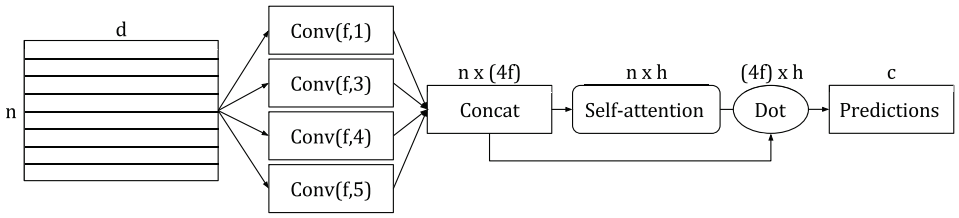

Figure 1: Scheme of SAS-MC-v1.

that $\phi(\mathcal{T}) = \mathbf{y}$, where $\mathcal{T} \in \{w_1, w_2, ..., w_n\}$ is a given textual synopsis with n words represented as vectors $\mathbf{w} \in \mathbb{R}^d$. We use pre-trained word vectors from (Pennington, Socher, and Manning 2014) and approximate the function $\phi(\cdot)$ by using convolutional layers and/or the self-attention mechanism, depending on the incarnation of SAS-MC. Note that $\mathbf{y} \in \{0, 1\}^c$ is a binary vector, in which each position represents one of the c labels (i.e., movie genres).

Inspired by (Wehrmann, Mattjie, and Barros 2018; Wehrmann et al. 2017a), the incarnations of SAS-MC are mostly shallow approaches that comprise at most a single layer of parallel convolutions for extracting features from synopses. Note that that the word-embedding itself already carries high-level semantic information from the text, significantly differing from recent computer vision approaches (Szegedy et al. 2014; Huang et al. 2016; Wehrmann et al. 2017b) that require several stacks of convolutional layers to extract high-level features from raw pixels. Next, we detail the self-attention mechanism and two versions of the proposed approach, namely SAS-MC-[v1,v2].

2.1 Self-attention Mechanism

The self-attention mechanism is one of the core components within both variations of SAS-MC. It is used to explicitly encourage relevant information while suppressing irrelevant data. More specifically, it comprises a two-layer neural network that ultimately learns weights within $[0, 1]$ for the features at each time-step.

Let $H \in \mathbb{R}^{n \times z}$ be a dense representation of the textual input, where n denotes the number of time-steps (e.g., words), and z the size of the feature vectors (depending on the incarnation, z can be the dimension resulting from a convolutional layer or it can be equal do d, which is the dimension of the word-embedding). The first step of the attention mechanism consists in a fully-connected layer to reduce the dimensionality to p-dimensional feature vectors. Values are processed by a $tanh$ activation function (denoted by ζ) to project values into the $[-1, 1]$ range, generating $V \in \mathbb{R}^{n \times p}$.

$$V = \zeta(HW_1) \tag{1}$$

Following, we use an additional fully-connected layer with h neurons, generating an output of size $n \times h$. Softmax is used to ensure that the weighted sum of all features in each temporal step results in 1. The resulting weight map is denoted by $A \in \mathbb{R}^{n \times h}$. Therefore, A can be seen as a matrix that carries the importance of each time-step in h different viewpoints (or *hops*).

$$A = softmax(VW_2) \tag{2}$$

Finally, the weighted feature map $M \in \mathbb{R}^{z \times h}$ is given by $M = H^T A$.

2.2 SAS-MC-v1

Our first approach employs four convolutional layers applied directly over word-embeddings (see Figure 1). SAS-MC-v1 is a somewhat modified version of the architecture introduced in (Kim 2014). In summary, SAS-MC-v1 convolves the input words with distinct parallel convolutional layers with different filter sizes f. This approach allows learning multiple n-gram-like features, where n depends on the value of f.

SAS-MC employs four convolutional layers with filter sizes $f \in \{1, 3, 4, 5\}$. We zero-pad the input matrix $\mathcal{T}$ in order to keep the resulting size unchanged. This is particularly important so that one can concatenate all generated feature maps into a single matrix that contains all the features extracted from the text. Note that the convolutional layer with $f = 1$ plays an important role within SAS-MC: it allows the self-attention to individually select the most important words for classifying a given synopsis, rather than limiting the architecture to learn from temporal data alone, as in (Kim 2014).

Let $\psi(\mathcal{T}) = \mathbf{X}$ be the computation of a convolutional layer applied over the input $\mathcal{T}$, which generates a feature map $\mathbf{X} \in \mathbb{R}^{n \times f}$. The final feature representation, which feeds the self-attention mechanism, is denoted by $\hat{\mathbf{X}} \in \mathbb{R}^{n \times 4f}$, given that we concatenate the resulting feature maps from the four convolutional layers.

Assuming that matrix $\hat{\mathbf{X}}$ contains all the information needed for classifying a given synopsis, we let the self-attention scheme to highlight the most important features while suppressing the irrelevant ones. This step generates a $f \times h$ matrix, which is flattened into a vector. Such a vector feeds the a fully-connected layer activated by c logistic sigmoid neurons, which show per-class probabilities $\hat{\mathbf{y}}$.

2.3 SAS-MC-v2

Figure 2 depicts the overall structure of SAS-MC-v2. Similarly to (Joulin et al. 2016), we design SAS-MC-v2 assuming that a word-vector itself already provides all information needed for understanding a given text. Therefore, this approach employs the self-attention mechanism directly over

the word-embeddings $\mathcal{T}$. Even though this strategy constraints the amount of temporal information, it helps to reduce the model complexity by reducing the number of parameters. In addition, this version is significantly faster than SAS-MC-v1.

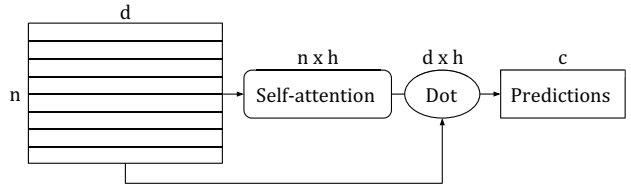

Figure 2: Scheme of SAS-MC-v2.

2.4 Loss Function

For training the models, SAS-MC optimizes a multi-label loss function – the binary cross-entropy for multiple classes, given by

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{i=1}^{c} [\mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i)] \quad (3)$$

where, c is the number of classes (i.e, number of movie genres), $\mathbf{y}_i$ is the actual class, and $\hat{\mathbf{y}}_i$ is the probability predicted by SAS-MC that the movie plot belongs to the i^{th} class. Note that y is a binary vector of length c and $\hat{y} \in \mathbb{R}^c$.

3 Experimental Setup

In this section, we detail the dataset that is used in our experiments (Section 3.1), as well as the evaluation criteria that we employ to measure predictive performance (Section 3.2), the baseline algorithms (Section 3.3) that we use to compare with SAS-MC, and the hyper-parameter settings for SAS-MC (Section 3.4).

3.1 Dataset

We validate the performance of our method by using the recently introduced dataset LMTD (Labelled Movie Trailer Dataset) (Simões et al. 2016; Wehrmann and Barros 2017a). LMTD comprises about 10,000 movie trailers from 22 genres, which were assigned according to the IMDB meta-data. In addition, it comprises roughly 400 hours of video and $\approx$ 30 million frames. To the best of our knowledge, this is the only dataset to provide a benchmark for multi-label classification of movie genres, namely LMTD-9. Such subset provides $\approx 4,000$ movie trailers annotated in 9 non-disjoint genres: namely action, adventure, comedy, crime, drama, horror, romance, sci-fi, and thriller. Those genres were selected as a consequence of limiting each class to have at least 10% of the total number of instances in LMTD. In LMTD-9, each instance is assigned to at least 1 and at most 3 genres.

Note that LMTD-9 is designed to multi-label learning based on visual data of movie trailers, while we propose a method for learning from textual synopsis. Hence, we make use of the LMTD metadata to retrieve the synopsis for each movie. In addition, whereas the authors of LMTD provide original train and test splits, we choose to adopt a k-fold cross-validation strategy to mitigate the results variance caused by the unbalanced classes, despite being particularly helpful when the amount of available data is limited. For reporting the final results, we concatenate predictions from the k test sets to compute unique per-class Area Under Precision-Recall Curves. Per-class results are then combined by different average methods to obtain a single estimation (see details in Section 3.2). We use five-fold cross-validation to train, evaluate, and test all methods.

Table 1: LMTD-9 dataset.

Genre	#Movies	Syn. Mean Length	Syn. Max Length
Action	856	27.28	68
Adventure	593	28.20	68
Comedy	1562	26.50	67
Crime	659	26.30	62
Drama	2032	26.67	70
Horror	436	26.78	60
Romance	651	26.72	70
SciFi	313	27.63	64
Thriller	693	26.73	64
1 genre	1264	26.37	66
2 genres	1740	26.96	70
3 genres	1017	26.90	64
Total Movies	4021	26.76	70

3.2 Evaluation Measures

Given that SAS-MC generates per-class probability values, and the same is true for the baseline algorithms, we follow the trend of multi-label classification research in which we avoid choosing thresholds by employing precision-recall curves (PR-curves) as the evaluation criterion for comparing the different approaches. For generating a PR-curve for a given classification method, one must select a predefined number of thresholds within $[0, 1]$ to be applied over the outputs of each method, finally generating several precision and recall points in the PR plane. The interpolation of these points (Davis and Goadrich 2006) constitute a PR-curve, and the quantitative criterion one analyzes is the area under that curve ($AU(\overline{PRC})$).

We employ the following derived measures: $AU(\overline{PRC})$ (micro average), $\overline{AU(PRC)}$ (macro average) and $AU(\overline{PRC})_w$ (weighted average). Each of those measures allow understanding different aspects regarding the method's capabilities. For instance, $\overline{AU(PRC)}$ measure is calculated by averaging the areas of all classes, which causes less-frequent classes to have more influence in the results. $AU(\overline{PRC})$ is calculated by using all labels globally, providing information regarding the whole dataset, which makes high-frequency classes to affect more the results. Finally, $AU(\overline{PRC})_w$ is calculated by averaging the area under precision-recall curve per genre, weighting instances according to the class frequencies.

3.3 Baseline Algorithms

As far as we know, this the first study to evaluate multi-label movie genre classification by analyzing synopsis. Hence, we compare SAS-MC to traditional and state-of-the-art approaches for single-label text classification methods, namely LSTMs (Hochreiter and Schmidhuber 1997), GRUs (Chung et al. 2015), Textual Convolutions (henceforth simply *Conv*) (Kim 2014), and FastText (Joulin et al. 2016).

3.4 Hyper-parameters

We performed a non-exhaustive grid-search for finding proper hyper-parameter choices for our methods and for the LSTM and GRU approaches. For Conv (Kim 2014) and FastText (Joulin et al. 2016), we used the recommended hyper-parameters described in their original papers. LSTMs and GRUs were trained with 256 neurons and dropout of 0.5 in the readout layer. We trained SAS-MC-[v1,v2] by using $f = 110$, $h = 25$, $p = 100$, and dropout of 0.2.

All models were trained using the stochastic optimizer Adam (Kingma and Ba 2014) with learning rate ($\alpha = 1 \times 10^{-3}$ and mini-batches of 128 instances. Word-embeddings are pre-trained vectors from (Pennington, Socher, and Manning 2014) with $d = 300$. We train all models for a maximum of 100 epochs, early-stopping when the predictive performance in the validation does not improve for 10 consecutive epochs. On average, models from SAS-MC reach convergence within 10 epochs.

4 Experimental Analysis

In this section, we present the experimental results comparing the predictive performance of the following algorithms: GRU (Chung et al. 2015), LSTM (Hochreiter and Schmidhuber 1997), Conv (Kim 2014), and FastText (Joulin et al. 2016)

4.1 Quantitative results

In Table 2 we report the predictive performance of all methods. In addition, we evaluate the impact of updating the word-embeddings during training. Models trained without updating the word-embeddings are denoted as *Fixed*.

Note that both proposed methods outperform all baselines in all evaluation metrics (values in bold), with and without updating the pre-trained word vectors. This indicates that our approaches perform better for learning information from both frequent (values of $AU(\overline{PRC})$) and rare (values of $\overline{AU(PRC)}$) movie genres.

We noticed that all baselines performed better when not updating the word-embeddings. It seems that using the back-propagated gradients to update those embeddings may damage the pre-trained vector structure, leading to poor results. However, for our simpler approach, namely SAS-MC-v2, we find that such an update leads to a slightly better predictive performance. We believe that in the backward pass, the self-attention mechanism acts as a filter of gradients, so that important words may suffer larger updates, while gradients from irrelevant words are filtered out. For instance, the network could update an important word-embedding to make it easier to recognize a given genre when that word appears, while keeping the rest of the embedding unchanged.

Table 2: Results comparing SAS-MC with the state-of-the-art methods with fixed and trainable GloVe embedding vectors.

Method	$AU(\overline{PRC})$	$\overline{AU(PRC)}$	$AU(\overline{PRC})_w$
GRU (Fixed)	0.658	0.584	0.640
LSTM (Fixed)	0.655	0.582	0.636
CONV (Fixed)	0.651	0.582	0.635
FastText (Fixed)	0.652	0.574	0.631
GRU	0.640	0.572	0.627
LSTM	0.634	0.563	0.618
CONV	0.656	0.582	0.639
FastText	0.575	0.495	0.559
SAS-MC-v1 (Fixed)	0.665	0.592	0.647
SAS-MC-v2 (Fixed)	0.669	0.606	0.654
SAS-MC-v1	0.660	0.595	0.646
SAS-MC-v2	**0.674**	**0.610**	**0.658**

It is interesting to notice that SAS-MC-v2 outperform traditional state-of-the-art approaches despite being conceptually much simpler. It provides a relative improvement of $\approx 3\%$ $\overline{AU(PRC)}$ over the strongest baseline, namely GRU. In general, RNNs (Recurrent Neural Networks) excel at learning temporal information, but at a high cost in terms of both model complexity (e.g., amount of parameters) and speed. Even though our approaches are not designed to learn long-term temporal dependencies, they provide structured vector representations of movie synopsis while running two to three orders of magnitude faster than the RNN baselines. In addition, such vectors proved to be much more suitable to learn movie genres in a multi-label fashion.

Observe that the self-attention mechanism seems to be more helpful when connected directly to the word-embeddings (SAS-MC-v2) rather than to temporal segments processed by convolutions (SAS-MC-v1). For instance, the performance of SAS-MC-v1 (Fixed) shows the impact of self-attention in a convolutional structure, which generated an absolute improvement in $AU(\overline{PRC})$ of only 1.0×10^{-2} when compared to CONV (Fixed). On the other hand, SAS-MC-v2 improved $AU(\overline{PRC})$ results of FastText (Fixed) in 3.2×10^{-2}. This improvement is $11.5\times^{-2}$ (0.606 *vs* 0.495) when comparing SAS-MC-v2 and FastText models trained with updates over the embedding.

Table 3 depicts the performance for all nine classes of the dataset. Once again, our approaches presented sound performance. SAS-MC-v1 outperformed all baselines in all genres, and SAS-MC-v2 provided solid improvements over its direct competitors. As expected, all methods tend to present better results for high-frequency classes (e.g., Drama and Comedy). Note that FastText underperforms in rare and subjective genres (e.g., 0.325 of AUPRC in SciFi), while our approaches present much better results in those cases. Finally, we observe that SAS-MC-v1 allows better understanding of Adventure, Horror, SciFi, and Thriller when compared to Conv, which is a similar approach without self-attention.

Table 3: Results comparing SAS-MC with the state-of-the-art methods.

Genre	SAS-MC-v2	SAS-MC-v1	CONV	FastText	GRU	LSTM
Action	**0.689**	0.674	0.673	0.585	0.662	0.656
Adventure	**0.620**	0.599	0.576	0.465	0.600	0.577
Comedy	**0.747**	0.739	0.735	0.641	0.707	0.710
Crime	**0.662**	0.644	0.639	0.570	0.656	0.618
Drama	**0.765**	0.755	0.760	0.699	0.739	0.732
Horror	**0.587**	0.586	0.567	0.422	0.546	0.554
Romance	**0.484**	0.464	0.454	0.413	0.459	0.434
SciFi	**0.525**	0.484	0.449	0.325	0.402	0.404
Thriller	**0.404**	**0.405**	0.380	0.333	0.374	0.374

4.2 Visualizing the Attention Mechanism

Figures 3 and 4 depict the visualization of the self-attention mechanism trained in SAS-MC-v2. We randomly selected a set of movies' synopses in order to visualize the estimated importance of each word for generating a given prediction. Analyzing both figures, we conclude that our method can leverage weighted word information for providing accurate multi-label predictions.

5 Related Work

Related work approaches are mostly designed to perform single-label classification of movie trailers. Nevertheless, in real world scenarios, a movie rarely belongs exclusively to a single genre, and processing an entire movie trailer is costly. In addition, note that the work described in this section are not directly comparable to ours, given that: i) most of the work assume the problem to be single-label classification; and ii) they make use of visual or audio data for training. Hence, comparing results from different modalities may be misleading and possibly unfair.

Rasheed et al. (Rasheed, Sheikh, and Shah 2005) propose the extraction of low-level features to detect movie genres through the application of the mean-shift classification algorithm (Comaniciu and Meer 2002). Such features are responsible for describing raw video elements, such as the average shot length, color variance, lighting key, and motion.

Another approach for movie genre classification makes use of well-known image descriptors to compute high-level features for each keyframe. The work of Zhou et al. (Zhou et al. 2010) employ the image descriptors Gist (Oliva and Torralba 2001), CENTRIST (Wu and Rehg 2008), and w-CENTRIST to extract high-level features from frames and then perform genre classification via the k-NN algorithm.

Huang and Wang (Huang and Wang 2012) propose a hybrid approach that combines both low-level visual features and audio information, resulting in a total of 277 features. They extract audio features such as audio intensity (measured in terms of the the RMS amplitude), timbre (based on different structures of amplitude spectrum), and rhythm.

Simões et al. (Simões et al. 2016) propose to make use of a ConvNet for extracting visual high-level features from movie frames. The ConvNet is trained at frame level in the LMTD-4 dataset proposed by the authors themselves.

Similarly, Wehrmann et al. (Wehrmann et al. 2016) propose the use of five neural networks (4 ConvNets and 1 MLP) to learn different aspects from the movie trailers. In their approach, 3 GoogleNets (Szegedy et al. 2014), 1 C3D (Tran et al. 2015), and 1 MLP form an ensemble of networks. An SVM is employed to perform the final genre classification by using several class predictions as features.

Wehrmann and Barros (Wehrmann and Barros 2017b) introduced LMTD-9, the first dataset for training and benchmark of multi-label movie genre classification models. This work provides a framework for learning multi-label genres by performing Convolutions-through-time (CTT) over deep visual features extracted from a Convolutional Network. That work has been extended in (Wehrmann and Barros 2017a), when the authors tested CTT in a two-network architecture that is capable of learning from both visual and audio data.

6 Conclusion

This paper proposed SAS-MC, a self-attentive method for synopsis-based multi-label movie genre classification. It comprises a self-attention mechanism to automatically encourage features from important time-steps. We provide two main approaches that connect this mechanism to convolutional layers or directly to the word-embedding input. We have shown that SAS-MC comfortably outperforms the best current approaches for all genres in a large movie trailer dataset, establishing itself as the novel state-of-the-art for synopsis-based multi-label movie genre classification. As future work, we want to provide hybrid models that can leverage from a multimodal self-attention mechanism. In addition, we intend to evaluate the proposed approaches in other similar tasks.

7 Acknowledgments

We would like to thank Google, Motorola and the Brazilian research agencies CAPES, CNPq, and FAPERGS for funding this research. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

References

Chung, J.; Gülçehre, Ç.; Cho, K.; and Bengio, Y. 2015. Gated feedback recurrent neural networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2067–2075.

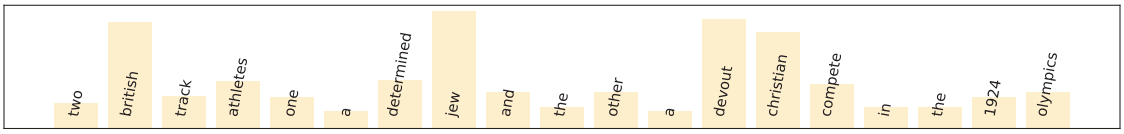

Figure 3: Visualization of the self-attention mechanism. Movie: Chariots of Fire (1981). Correct genres: Drama. Predicted genres (top-3): Drama (85%), Romance (35%), Comedy (28%).

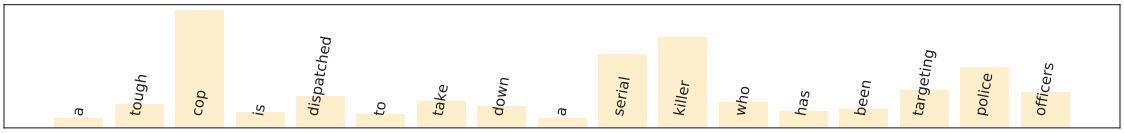

Figure 4: Visualization of the self-attention mechanism. Movie: Blitz (2011). Correct genres: Action, Crime, Thriller. Predicted genres (top-3): Crime (94%), Thriller (78%), Action (70%).

Comaniciu, D., and Meer, P. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5):603–619.

Davis, J., and Goadrich, M. 2006. The relationship between precision-recall and roc curves. In *ICML*.

Goodfellow, I. J.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. The MIT Press.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780.

Huang, Y.-F., and Wang, S.-H. 2012. Movie genre classification using svm with audio and video features. In Huang, R.; Ghorbani, A. A.; Pasi, G.; Yamaguchi, T.; Yen, N. Y.; and Jin, B., eds., *AMT*, volume 7669 of *Lecture Notes in Computer Science*, 1–10. Springer.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision* 42(3):145–175.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rasheed, Z.; Sheikh, Y.; and Shah, M. 2005. On the use of computable features for film classification. *Trans. on Circuits and Systems for Video Technology* 15(1):52–64.

Simões, G.; Wehrmann, J.; Barros, R. C.; and Ruiz, D. D. 2016. Movie genre classification with convolutional neural networks. In *IJCNN*, 8. IEEE.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2014. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *2015 ICCV*, 4489–4497. IEEE.

Wehrmann, J., and Barros, R. C. 2017a. Movie genre classification: A multi-label approach based on convolutions through time. *Applied Soft Computing*.

Wehrmann, J., and Barros, R. C. 2017b. Convolutions through time for multi-label movie genre classification. In *ACM Symposium on Applied Computing*, 6. ACM.

Wehrmann, J.; Barros, R. C.; Simões, G.; Paula, T. S.; and Ruiz, D. D. 2016. (Deep) Learning from Frames. In *Brazilian Conference on Intelligent Systems*, 6.

Wehrmann, J.; Becker, W.; Cagnini, H. E.; and Barros, R. C. 2017a. A character-based convolutional neural network for language-agnostic twitter sentiment analysis. In *IJCNN*, 2384–2391.

Wehrmann, J.; Simões, G. S.; Barros, R. C.; and Cavalcante, V. F. 2017b. Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing*.

Wehrmann, J.; Mattjie, A.; and Barros, R. C. 2018. Order embeddings and character-level convolutions for multimodal alignment. *Pattern Recognition Letters* 102:15 – 22.

Wu, J., and Rehg, J. M. 2008. Where am i: Place instance and category recognition using spatial pact. In *CVPR*, 1–8.

Zhou, H.; Hermans, T.; Karandikar, A. V.; and Rehg, J. M. 2010. Movie genre classification via scene categorization. In *Proceedings of the international conference on Multimedia*, 747–750. ACM.