Improving Trope Detection with Crowdsourced Examples

Stanford CS224N Custom Project

Jean-Peïc Chou
Department of Computer Science
Stanford University
jeanpeic@stanford.edu

Abstract

Tropes are cultural narrative conventions that shape our expectations of stories. They thoughtfully encompass our understanding of the world and the representation of our culture. The use and comprehension of tropes require advanced human language skills such as causal, motivational, or pragmatic inferences. Detecting them is still an open and challenging research question that might be key to build 'intelligent' machines. In this regard, we propose to leverage the knowledge gathered by thousands of passionate people about tropes on *tvtropes.org*. Our contribution is twofold. First, we provide a dataset of specific examples for each trope extracted from the tropes wiki. From this scraped data, we propose the original task of the classification of tropes' examples. It offers the main advantages of being more workable and interpretable due to the concision and relevance of the examples annotated by the community of tropers. Second, we present attempts to tackle this new challenge based on a word and sentence-level analysis and show the potential applications of such models on the more general task of detecting tropes in larger texts such as synopses. The proposed models outperform the state-of-the-art network when trained on individual tropes by making use of different levels of cues (word, sentence, or full context) based on the nature of the trope studied. This confirms the need to adapt the detection method or to build a multi-level comprehension model.

1 Key Information to include

- Mentor: Anna Goldie
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

Stories are the pillars of human cultures. They shape our imagination, forge our understanding of the world, and ensure our cultural heritage. As such, the exploration of stories has been fascinating many philosophers and scientists since the beginning of the 20th century. Russian formalist Vladimir Propp was one of the firsts to systematically attempt to break down tales in a few categories [1]. Many other derived models and substantial story grammars were later conceived to decompose stories. However, most of them have been discredited because of the finiteness of the elements in their lexicon [2], because of their high and intricate abstraction in computational contexts, or their lack of data and examples.

Overcoming these deficiencies, a whole vocabulary of narrative units called tropes has been imagined and inventoried by a community of thousands of enthusiasts on the website *tvtropes.org*. Tropes are

narrative devices or storytelling conventions. They are recognizable patterns found in all kinds of media. Tropes can describe every level of a wok: the story and its discourse, characters and their interactions, location, time... For instance, the *Save the Princess* trope depicts the universal story plot in which a character, often portrayed by the *Damsel in Distress* trope, is kidnapped, and later rescued by the *Hero*. Tropes are highly sophisticated as they can hold abstract concepts related to morals, behaviors, or motivations. In this regard, the study of tropes can give access to human representations of the world and implicit knowledge they carry.

In this paper, we will tackle the challenge of trope detection in movie synopses introduced by Chen-Hsi Chang et al. [3]. We will work on the dataset TiMoS, containing 5,623 movie synopses as well as the trope appearances in each of the movies for 95 hand-picked tropes. In their work, the authors note the extreme difficulty of this task, requiring advanced linguistic, cognitive, and social skills, even for humans. A system able to achieve good results on this challenge could better tackle real world tasks such as recommendation systems, chat-bot, or opinion mining. With this dataset, the authors proposed MulCom, a multi-level comprehension network conceived for this task. If MulCom's performances are very limited as the authors admit, it opens many avenues for improvement. In this regard, we propose to exploit external examples scraped on the website *tvtropes.org* where media works were thoroughly annotated, and preprocessed them to obtain more than 560,000 examples for 89 tropes. We then introduce the new closely related task of trope identification based on examples. Adapting binary classifiers based on word-level features and trained on this task to TiMoS notably yields largely better results than MulCom. Eventually, we propose a derived model of MulCom directly applicable to TiMoS and some leads to harness examples in this task.

3 Related Work

Stories have largely been exploited to assess machine comprehension performances. More specifically, movie synopses were analyzed through the lens of tags created by Sudipta Kar et al. in order to classify or compare them, as well as analyze their flow of emotions [4]. However, these tags remain quite shallow compared to the rich and dense library of tropes. Another way to probe movie synopses was proposed by Tapaswi et al. who introduced MovieQA [5], a dataset of plot synopses and multiple-choice questions aiming at evaluating story comprehension. Unfortunately, MovieQA's possible answers were shown to be biased by B. Jasani et al. who demonstrated that the plot synopsis was not necessary to achieve satisfying results [6].

As of today, publications mentioning tropes are quite rare. In their study in trailer generation, Smith et al. were one of the firsts to cast light on tropes, describing weak tropes as visual elements characterizing genres [7]. Tropes have mostly been used as fuel for story generation [8] or social analyses [9]. They have only recently been considered as a vocabulary for advanced story grammars. In this regard, the website *tvtropes* was thoroughly examined [10, 11]. Its structure was shown to provide a thoughtful backbone for such a grammar as well as rich information about tropes interactions. Trope detection was mentioned a few times but was tackled first by Chen-Hsi Chang et al. [3]. They conceived a Multi-Level Comprehension Network (MulCom) backed by a Multi-Step Recurrent Relational Network (MSRRN). The model aims at combining word-level information from Word2Vec, sentence-level information from a BERT encoder coupled with a RNN, and relation-level information from their MSRRN, a GNN based on characters' interactions. Although the network architecture was built to be interpreted quite naturally, the role of some layers and operations remains unclear in the learning process. Besides, the authors note that even human evaluators were unable to detect some of the tropes when relying on the synopses alone. With commonsense inferences, the evaluators increased their final score by 30%, showing that additional cultural facts about stories and tropes are crucial in this task.

4 Approach

Trope Identification. In the first part of our work, we collected examples for each of the studied tropes by analyzing the structure of the website. Based on this new dataset, we propose the new original task of identifying the trope with examples as input. In practice, this task can be handled in two different ways.

(i) This can be considered as a multi-class classification problem (MC). For each example, the model needs to decide which trope it is an example. This is the most natural way of using the dataset. However, in practice, sentences can be cues for several tropes at the same time. For instance, the *Would Hurt a Child* trope that characterizes someone who is so evil that they might hurt a defenseless person is often found with the *Big Bad* trope which corresponds to the main villain in a plot. (ii) Therefore, we can divide this task into as many binary classification tasks as there are tropes studied. The problem can then be considered as a multi-label classification one.

The perks of this challenge over the more global trope detection in synopses are that it is more workable and interpretable. Indeed, examples from *tvtropes* are generally a few sentences long and straightforward whereas synopses generally have 50 to 100 sentences and require wild guesses based on commonsense to detect some of the tropes.

To tackle the MC and ML tasks, we use and compare 3 different models on the dataset:

- A Support Vector Machine (SVM) trained on TF-IDF features
- A neural network with two BiLSTM layers followed by one dense layer using pretrained word vectors from GloVe as input [12]
- The pretrained encoder BERT [13] with a classification head on top

Once these models are trained, we want to apply them to TiMoS whose texts' nature are different from the crowdsourced examples. Based on the models trained for the MC and ML task:

- SS (Sentence by Sentence): We predict the probabilities of each class (trope) for each sentence of the synopses. For each class, we choose the final predicted probability as the maximum one over all sentences.
- FT (Full Text): We predict the probabilities of each class on the whole synopses.

The best threshold of the probability for each class if found on the training set of TiMoS.

Trope Detection. In this second part, we propose a derived model from MulCom [3] aiming at removing some of its layers and operations and at making it more straightforward. We will call it MulCom2.

The task is a multi-label classification problem. Given a synopsis, the model predicts the appearance of each trope in it. The output is a binary vector $T = t_1, t_2, \dots$ where t_i corresponds to the appearance of trope i and $|T|$ is the number of tropes studied. Following the intuition of authors of MulCom, we want to combine different levels of cues in order to detect tropes. Indeed, some tropes depend on single sentences or even words such as *Would Hurt a Child* whereas *Big Bad* might need more hints about a character throughout the story. The model is divided into 3 similar main parts: *Knowledge*, *Comprehension*, *Classification*.

Knowledge. The model stores a learnable embedding matrix representing the embeddings of tropes $E \in \mathbb{R}^{|T|*d_T}$ where T is the trope set and d_T is the dimension of the embeddings. E is randomly initialized and is used to as queries to perform attention without needing prior knowledge.

Comprehension. The model extracts information from the text with different level of comprehension in each stream. For each of them, it uses tropes knowledge as queries of a multiplicative attention mechanism to keep the most valuable information.

- At the word-level, keys and values are word vectors from the pretrained GloVe model.
- At the sentence-level, keys and values are sentence embedded by the encoder BERT.
- At the paragraph-level, keys and values are paragraphs with the maximum length embedded by the encoder BERT (512 tokens).

Classification. In this last step, the output of each stream is concatenated for each trope to obtain the output matrix $M \in \mathbb{R}^{|T|*(d_{GloVe}+d_{BERT}+d_{BERT})}$. It is then sent to a linear classifier that will give the final prediction. We use Binary Cross Entropy as our loss function to train on:

$$\frac{1}{B}\sum_{b=1}^{B}\frac{1}{T}\sum_{t=1}^{T} -[y_{b,t} \cdot log(\hat{y}_{b,t}) + (1 - y_{b,t}) \cdot log(1 - \hat{y}_{b,t})]$$

where B is the batch, $y_{b,t}$ and $\hat{y}_{b,t}$ are the ground truth and prediction of the model of trope t in batch b.

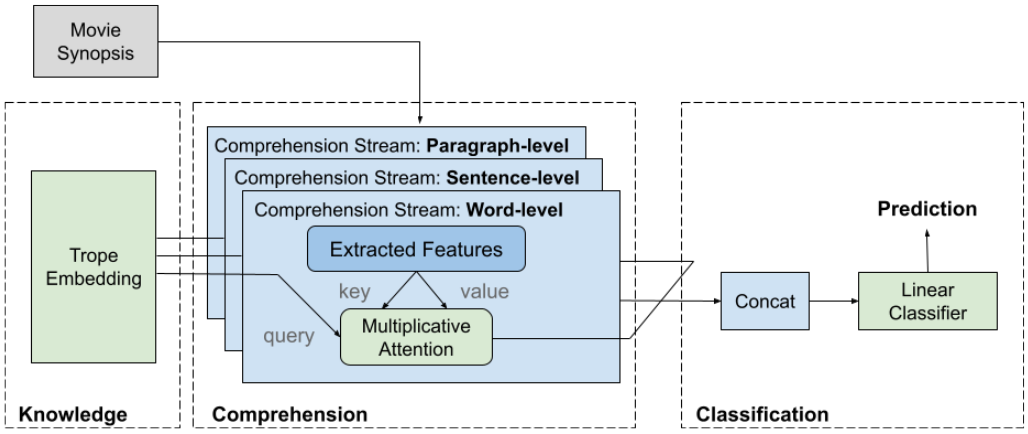

Figure 1: MulCom2 is designed after MulCom [3] to deal with different levels of cues in movie synopses with multiple streams.

5 Experiments

5.1 Data

From the wiki *tvtropes*, we extracted over 760,000 examples from films, character pages, books, anime, comic books, manga, and TV shows for 89 tropes. Each example (text) is associated with exactly one trope (class/label). The scraped dataset was split into train, validation, and test sets.

TiMoS analyzed 93 tropes whereas we were only able to extract examples for 89 tropes. This is due to the fact that some tropes have been removed, renamed, or subdivided into multiple tropes. The results of our work will thus not be exactly equivalent. We also note that the level of information provided by each scraped example is very inconsistent. Some don't give enough context (e.g. "*Big Bad*: Voldemort") or on the contrary too much (e.g. by giving hundreds of inextricable lines of dialogue). In addition, the level of abstraction of the examples varies a lot making the dataset hard to read from the model's perspective (e.g. "Voldemort is the main antagonist", versus why or how he is, versus the description of a specific scene that depicts him like that...). We curated examples that we deemed appropriate for the model to learn properly by culling examples containing less than a certain number of words or examples that were incomprehensible without their context. The word clouds in Figure 2 demonstrate the relevance of the remaining examples. The *Would Hurt a Child* trope word cloud is mainly composed by the words "children", "killing", "try" or "threaten" for instance. Naturally, the *Alcoholic* character trope is described with alcohol-related words such as "drunk", "alcohol', or "liquor".

Figure 2: Word clouds of the examples of *Would Hurt a Child* (left) and *The Alcoholic* (right)

We are left with about 560,000 examples with a median of 6,032 examples per trope. The minimum number of examples is 1,003 (Cluster-F Bomb trope) while the maximum is 25,165 (Foreshadowing trope). Classes are therefore very imbalanced. For MC and ML task, we balance them by randomly sampling each class based on the minimum number of tropes.

For the individual binary classification tasks (ML), we also randomly sample the same number of tropes than the trope studied for each of them. In addition, we create for each trope a *hard* dataset (HD) that only comprises examples of tropes which are close to the trope studied. The closeness of two tropes is assessed based on the similarities of their word frequencies. Figure 3 shows a t-SNE visualization of tropes relying on the GloVe embedding of their 100 most frequent words. We can observe that *Bittersweet Ending* and *Earn Your Happy Ending* are close as they both relate to the end of a story (in blue on the left). Same thing goes for *Big Bad* and *The Dragon* (the *Big Bad*'s right hand) for instance (in read toward the middle).

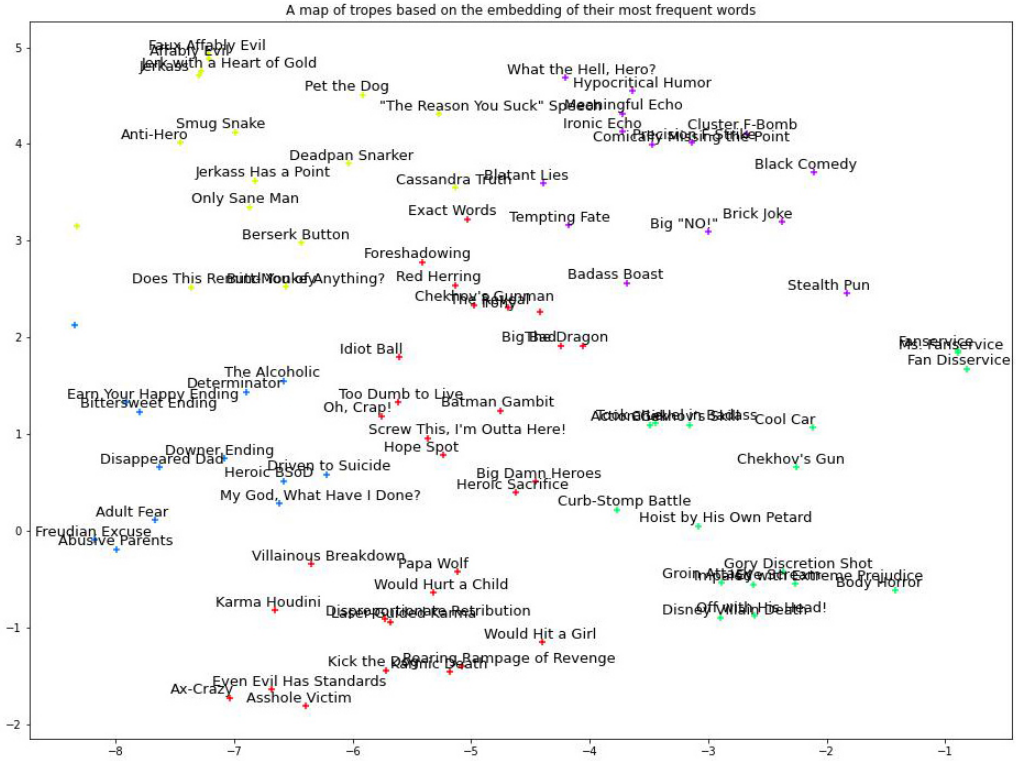

Figure 3: t-SNE visualization of tropes based on the embedding of their frequent words.

5.2 Evaluation method

Trope Identification. The performances on the MC and ML tasks are evaluated with F1 and Area Under the ROC Curve (AUC) scores to determine how well the models can distinguish the classes no matter the threshold.

Models trained on MC and ML tasks are mainly compared to the best and worst performance of MulCom on individual tropes, i.e. respectively the *Chekhov's Gun* trope and the *Would Hurt a Child* trope.

Trope Detection. On TiMoS test set, the performance is based on the F1 and mean average precision (mAP) metrics used by Chen-His Chang et al. [3] in order to compare our models with MulCom and the other baselines provided by the authors (Random, BERT [13] fine-tuned and Folksonomication [4]). In addition, We compare the models with the strategy of always detecting every trope as another baseline.

5.3 Experimental details

For MC and ML tasks, SVM and BiLSTM classifiers would take seconds to minutes to train and predict. For the first model, we tried several classifiers on tf-idf features (xgboost, random forest, MLP...) but SVM is the one that gave us the best result. For the second model, we tested different sizes of hidden layers and finally kept 64 for all of them. We also tried various pretrained word vectors such as Google's Word2Vec or tried training it ourselves, but GloVe was the best model. We used Hugging face implementation of BERT pretrained model on uncased vocabulary for the last model. For each individual trope classifier, finetuning the full model would take 1 to 2 hours based on the evolution of the validation loss. For the MC version, the training runtime was around 12 hours as the dataset was much larger. We used batches of 16 examples over 3 epochs and stopped the training early if loss was not going in the right direction when evaluated. The learning rate was fixed to $2e-5$.

The adaptation of the MC and ML models to TiMoS could be achieved in a lot of different ways. Instead of keeping the maximum probability predicted over all sentences, we tried using the whole average or the average of the N best (which would make sense for certain tropes). We tried also working by batch of sentences and by only considering some part of the synopses. The selected methods are the ones that gave the best results.

MulCom2's trope embedding size was determined after a few tries and fixed at 64. In order to leverage the additional examples with this model, the goal was to use previously fine-tuned BERT model for the MC task or train it on TiMoS as well as the crowdsourced examples. Unfortunately, we were short on time as each training would take up to 8 hours. The slowest part of MulCom2's process was the encoding of words by BERT which we would do externally (as BERT is only used as a feature extractor and not fine-tuned). We would then directly train the network by feeding it the vector representation of the synopses rather than the texts. Lacking time, MulCom2 was only trained and tested on synopses from TiMoS which took over 11 hours. It was supposed to be pretrained and trained on the examples as well as on synopses augmented by trope examples which would have required a day of computing.

Eventually, the process of scraping data from all media on tvtropes took approximately 15-20 hours.

5.4 Results

Results of the classification of the Woud Hurt a Child trope on the balanced dataset of examples and the hard one are shown in Table 1. We observe that BERT model is the one that performs the best with a F1 score of 92.41 and AUC of against 87.33 for SVM and 85.27 for the BiLSTM network. However, when applied to TiMoS to detect the same trope, the SVM model surprisingly outperforms all the other models including state-of-the-art MulCom with a F1 score of 21.67 when applied to the full text against 12.36 for MulCom. For both BERT and SVM, predicting on the full text returns better results than sentence by sentence which is not intuitive based on the nature of the trope. Indeed, a sentence describing a child being hurt or threatened should be a sufficient cue. Otherwise, SVM on *hard* examples struggles a bit more than on the randomly balanced dataset but this method does not make it more robust as its performance on TiMoS drops a bit.

Examples	**SVM**		**BiLSTM**	**BERT**		**SVM HD**		
F1	87.33		85.27	**89.41**		83.21		
AUC	93.42		92.12	**97.7**		91.36		
TiMoS	**SVM SS**	**SVM FT**	**BiLSTM SS**	**BERT SS**	**BERT FT**	**SVM HD SS**	**SVM HD FT**	**MulCom** [3]
F1	16.91	**21.67**	15.62	16.9	18.35	14.71	20.54	12.36

Table 1: F1 and AUC Scores on the models trained on the dataset of examples for *Would Hurt a Child* (balanced and *hard* dataset) and F1 score on TiMoS

The results of the classification of the *Chekhov's Gun* trope on the balanced and hard dataset shown in Table 2 are similar with BERT outperforming the other models on the binary classification task for both metrics. Here, BERT gives the best results with a F1 score of 40.61 against 38.58 for MulCom. This time, predicting the trope appearance sentence by sentence gives the best result which

is surprising as well as *Chekhov's Gun* trope depicts "a plot device that is not significant until later in the story". A broader grasp of the context would therefore be needed to detect it. Finally, we observe the same behavior for the model trained on the *hard* dataset.

Examples	**SVM**		**BiLSTM**	**BERT**		**SVM HD**		
F1	84.46		74.62	**89.2**		74.81		
AUC	92.15		82.2	**94.36**		83.18		
TiMoS	**SVM SS**	**SVM FT**	**BiLSTM SS**	**BERT SS**	**BERT FT**	**SVM HD SS**	**SVM HD FT**	**MulCom** [3]
F1	39.93	39.35	37.95	**40.61**	37.53	39.16	38.36	38.58

Table 2: F1 and AUC Scores on the models trained on the dataset of examples for Chekhov's Gun (balanced and *hard* datasets) and F1 score on TiMoS

As for the results on the full list of tropes based on synopses shown in Table 3, BERT and SVM multi-class classifiers trained on examples are achieving correct results which are smaller but close to other baselines BERT fine-tuned and Follksonomication. We note here that our 3 proposed models are largely dominated by MulCom, especially for MulCom2 with a F1 score of 18.57 compared to 25.00 for its original counterpart.

The F1 score for each trope given by the multi-class models BERT and SVM are shown in Appendix for more details. While some tropes are barely detected, others such as *Impaled with Extreme Prejudice* or *Eye Scream* show very good results.

Baseline	F1	mAP
Random	13.97	8.14
Always Detect	14.75	8.09
BERT [13] (fine-tuned)	23.97	17.26
Folksonomication [4] FastText	22.53	16.35
MulCom [3]	**25.00**	**18.73**
Our models	F1	mAP
SVM (trained on examples)	21.02	14.69
BERT (trained on examples)	22.71	16.14
MulCom2 (trained on synopses)	18.57	13.3

Table 3: F1 and AUC Scores on the models trained on the dataset of examples for Chekhov's Gun (balanced and hard) and F1 score on TiMoS

6 Analysis

The results show that word-level features such as simple tf-idf based on relevant examples can be very valuable in the detection of tropes. We note that the models' performance varies a lot based on the trope detected and that one model does not dominate all the others. This makes sense as tropes describe different abstraction levels of a work. The detection of certain tropes significantly depends on the appearance of specific words such as *Would Hurt a Child* or *The Alcoholic*, while others require the full context such as *Big Bad* or the *Chekhov's Gun*. Chen-Hsi et al. already noted that trope detection will need knowledge adaptation [3] depending on the trope by using different models. These results are overall a strong confirmation of this assessment.

The best results found by predicting sentence by sentence or on the full text for *Would Hurt a Child* and *Chekhov's Gun* are surprising. For the first trope, the reasons could be that the synopses are not precise enough and the model needs to find cues from different parts of the full text, or that the model is easily tricked by specific words in some sentences as it only relies on tf-idf features. As for *Chekhov's Gun*, if the plot device works at a high level, cues might be very specific and be found in certain sentences.

Some tropes such as *Irony* or *Hypocritical Humor* demand a deeper understanding of languages that might be too challenging for multi-class models. For all these quite abstract tropes or many other specific tropes, none of our models perform well, giving F1-scores rather low (down to 9.25 and 10.15 for *Exact Words*) compared to the worst F1-scores of MulCom (12.36) as observed in Figure 5 in Appendix. This important disparity of results explains why the overall F1 and mAP scores are lower than MulCom's one. Overall, it should be remembered that these multi-class models were only trained on external examples having therefore no knowledge about the actual structure of synopses and their cues to certain tropes which is remarkable.

Finally, MulCom2 did not yield the expected results which is explainable by the fact that it is very rapidly overfitting due to its relatively high number of parameters compared to the number of synopses it was trained on as we can see if Figure 4. As it might need more data, the use of the additional examples for pretraining and training as it was planned might give better results. More importantly, MulCom2 does not extract character interaction features (out of the scope of this study) which the original MulCom heavily relies on according to its results [3].

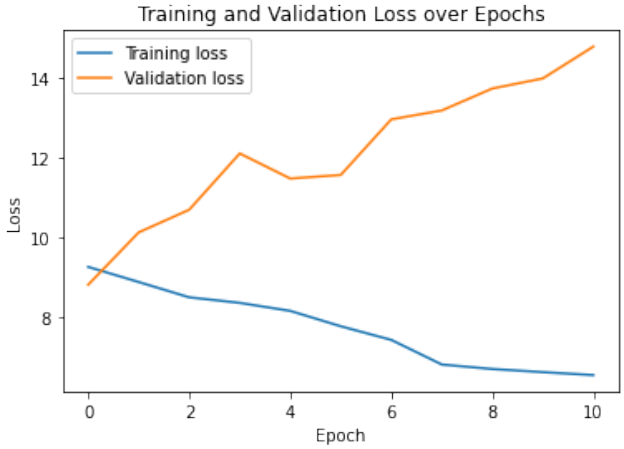

Figure 4: The validation loss of MulCom2 keeps increasing after only 1 epoch while the model keeps overfitting the training set.

7 Conclusion

In this project, we have collected and preprocessed a dataset of more than 560,000 precise examples for 89 tropes that we make available for future research on tropes. We showed that this dataset could be leveraged to achieve satisfying results on the trope detection in movie synopses challenge by training classifiers on the original task of trope identification alone. In addition, we demonstrated the ability of these models to beat the state-of-the-art model when trained as binary classifier on single tropes rather than as multi-label classifiers. Eventually, we proposed a derived model of the state-of-the-art multi-level comprehension network which is a simpler and more straightforward model that still needs some adjustments.

We believe this work opens many avenues of research on tropes. This project is far from assessing the full potential of these examples in the task of trope detection. One lead could be to analyze these examples in their context by probing all the other trope examples, synopses, and metadata from the original work they come from. Furthermore, classifiers trained on trope identification are more interpretable than the other proposed networks for trope detection as they enable to break down synopses and give the location of the cues it found in the text. Such findings could be a crucial property to understand how tropes are sequenced, how they interact, or in which part of the stories they mostly intervene, which could lead to the foundations of a tropes grammar. Among many other applications, this new rich dataset could also be used to create stories by asking a decoder model to generate text based on tropes chosen by a user. Eventually, these leads might also be combined with the tropes knowledge graph extracted from *tvtropes*' structure [11].

References

[1] Vladimir Iakovlevich Propp. *Morphology of the folktale*. University of Texas Press, 1968.

[2] Alan Garnham. What's wrong with story grammars. *Cognition*, 15(1):145–154, 1983.

[3] Chen-Hsi Chang, Hung-Ting Su, Jui-Heng Hsu, Yu-Siang Wang, Yu-Cheng Chang, Zhe Yu Liu, Ya-Liang Chang, Wen-Feng Cheng, Ke-Jyun Wang, and Winston H. Hsu. *Situation and Behavior Understanding by Trope Detection on Films*, page 3188–3198. Association for Computing Machinery, New York, NY, USA, 2021.

[4] Sudipta Kar, Suraj Maharjan, and Thamar Solorio. Folksonomication: Predicting tags for movies from plot synopses using emotion flow encoded neural network. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2879–2891, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[5] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Bhavan Jasani, Rohit Girdhar, and Deva Ramanan. Are we asking the right questions in movieqa? pages 1879–1882, 10 2019.

[7] John R. Smith, Dhiraj Joshi, Benoit Huet, Winston Hsu, and Jozef Cota. Harnessing a.i. for augmenting creativity: Application to movie trailer creation. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1799–1808, New York, NY, USA, 2017. Association for Computing Machinery.

[8] Andrea Guarneri, Laura Anna Ripamonti, Francesco Tissoni, Marco Trubian, Dario Maggiorini, and Davide Gadia. Ghost: a ghost story-writer. *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter*, 2017.

[9] Stacey Svetlichnaya staceys. Trope propagation in the cultural space. 2011.

[10] Rubén Héctor García-Ortega, Pablo García-Sánchez, and Juan Julián Merelo Guervós. Tropes in films: an initial analysis. *CoRR*, abs/2006.05380, 2020.

[11] Jean-Peïc Chou and Marc Christie. Structures in tropes networks: Toward a formal story grammar. *Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)*, 2021.

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

A Appendix

	Trope	F1		
		Always Detect	SVM	BERT
0	Big Bad	0.29369183	0.29950495	0.340632603
1	Jerkass	0.257655755	0.305317324	0.30335097
2	Faux Affably Evil	0.125	0.162393162	0.222222222
3	Smug Snake	0.127128263	0.161137441	0.146478873
4	Abusive Parents	0.116438356	0.281407035	0.226086957
5	Would Hurt a Child	0.105625718	0.172757475	0.164473684
6	Action Girl	0.105625718	0.198198198	0.176100629
7	Reasonable Authority Figure	0.116438356	0.174242424	0.156097561
8	Papa Wolf	0.08806489	0.213483146	0.178082192
9	Deadpan Snarker	0.311156602	0.318072289	0.340486409
10	Determinator	0.109965636	0.182352941	0.220183486
11	Only Sane Man	0.112128146	0.123222749	0.162162162
12	Anti-Hero	0.081395349	0.209205021	0.145833333
13	Asshole Victim	0.220064725	0.278301887	0.268907563
14	Jerk with a Heart of Gold	0.204570185	0.220082531	0.23179792
15	Even Evil Has Standards	0.162583519	0.204793028	0.212765957
16	Affably Evil	0.125	0.176870748	0.181818182
17	Too Dumb to Live	0.206521739	0.256781193	0.238167939
18	Butt-Monkey	0.158482143	0.186948854	0.182879377
19	Ax-Crazy	0.174778761	0.247349823	0.256198347
20	Berserk Button	0.220064725	0.225705329	0.255941499
21	Ms. Fanservice	0.143982002	0.177419355	0.191419142
22	The Alcoholic	0.107798165	0.273504274	0.318471338
23	Disappeared Dad	0.099078341	0.134228188	0.22556391
24	Would Hit a Girl	0.074679113	0.104918033	0.109090909
25	Oh, Crap!	0.351648352	0.365541327	0.368831169
26	Driven to Suicide	0.158482143	0.351145038	0.434285714
27	Adult Fear	0.172757475	0.256410256	0.232804233
28	Heroic BSoD	0.150224215	0.183168317	0.204724409
29	Big "NO!"	0.127128263	0.150289017	0.173228346
30	Eye Scream	0.156424581	0.334975369	0.476190476
31	Gory Discretion Shot	0.133484163	0.223728814	0.226415094
32	Impaled with Extreme Prejudice	0.099078341	0.380952381	0.418604651
33	Off with His Head!	0.08806489	0.192513369	0.365384615
34	Disney Villain Death	0.072429907	0.189473684	0.254545455
35	"The Reason You Suck" Speech	0.154362416	0.191780822	0.219114219
36	Tempting Fate	0.122866894	0.131147541	0.193146417
37	Disproportionate Retribution	0.146067416	0.196808511	0.242038217
38	Badass Boast	0.131370328	0.202531646	0.186915888
39	Groin Attack	0.180815877	0.261363636	0.261780105
40	Roaring Rampage of Revenge	0.109965636	0.265306122	0.256410256
41	Big Damn Heroes	0.141891892	0.209459459	0.252252252
42	Heroic Sacrifice	0.152295633	0.25974026	0.307692308
43	Screw This, I'm Outta Here!	0.139797069	0.212058212	0.208053691
44	Kick the Dog	0.202614379	0.248847926	0.282945736
45	Pet the Dog	0.180815877	0.227160494	0.219560878
46	Villainous Breakdown	0.120728929	0.2	0.171270718
47	Precision F-Strike	0.158482143	0.218623482	0.200488998
48	Cluster F-Bomb	0.083623693	0.136363636	0.214285714
49	Jerkass Has a Point	0.13559322	0.184397163	0.233576642
50	Idiot Ball	0.141891892	0.170212766	0.19047619
51	Batman Gambit	0.092485549	0.176470588	0.220532319
52	The Dragon	0.114285714	0.218487395	0.203125
53	Cool Car	0.092485549	0.199233716	0.388059701
54	Body Horror	0.101265823	0.27972028	0.376470588
55	The Reveal	0.146067416	0.187878788	0.213541667
56	Curb-Stomp Battle	0.103448276	0.228571429	0.25477707
57	Cassandra Truth	0.099078341	0.147239264	0.142180095
58	Blatant Lies	0.114285714	0.130879346	0.129032258
59	Crapsack World	0.103448276	0.238095238	0.181818182
60	Comically Missing the Point	0.103448276	0.115384615	0.11631664
61	Fanservice	0.172757475	0.218085106	0.238596491
62	Fan Disservice	0.131370328	0.22962963	0.203389831
63	Brick Joke	0.208469055	0.218375499	0.23117338
64	Hypocritical Humor	0.13559322	0.139318885	0.147710487
65	Does This Remind You of Anything?	0.131370328	0.131782946	0.176470588
66	Black Comedy	0.118586089	0.142857143	0.228571429
67	Irony	0.125	0.133152174	0.147909968
68	Exact Words	0.092485549	0.101492537	0.103932584
69	Stealth Pun	0.099078341	0.131428571	0.121212121
70	Bittersweet Ending	0.291925466	0.307692308	0.316368638
71	Karma Houdini	0.216216216	0.258302583	0.250406504
72	Downer Ending	0.204570185	0.231404959	0.211640212
73	Laser-Guided Karma	0.127128263	0.193181818	0.19
74	Earn Your Happy Ending	0.122866894	0.15042735	0.18630137
75	Karmic Death	0.107798165	0.236024845	0.168831169
76	My God, What Have I Done?	0.154362416	0.199445983	0.204724409
77	What the Hell, Hero?	0.122866894	0.139931741	0.141025641
78	Hope Spot	0.137697517	0.233333333	0.181818182
79	Took a Level in Badass	0.096885813	0.123364486	0.133333333
80	Chekhov's Gun	0.369565217	0.389244558	0.381313131
81	Foreshadowing	0.359840954	0.369198312	0.438461538
82	Chekhov's Skill	0.129251701	0.187845304	0.228187919
83	Chekhov's Gunman	0.101265823	0.165605096	0.150485437
84	Red Herring	0.103448276	0.195348837	0.181818182
85	Ironic Echo	0.188803513	0.230031949	0.211812627
86	Hoist by His Own Petard	0.148148148	0.21875	0.24691358
87	Meaningful Echo	0.092485549	0.109452736	0.172043011
88	Freudian Excuse	0.107798165	0.173228346	0.140186916

Figure 5: Results of SVM and BERT muti-class models