# Folksonomication: Predicting Tags for Movies from Plot Synopses Using Emotion Flow Encoded Neural Network

**Sudipta Kar** **Suraj Maharjan** **Thamar Solorio**
Department of Computer Science
University of Houston
Houston, TX 77204-3010
{skar3, smaharjan2, tsolorio}@uh.edu

## Abstract

Folksonomy of movies covers a wide range of heterogeneous information about movies, like the genre, plot structure, visual experiences, soundtracks, metadata, and emotional experiences from watching a movie. Being able to automatically generate or predict tags for movies can help recommendation engines improve retrieval of similar movies, and help viewers know what to expect from a movie in advance. In this work, we explore the problem of creating tags for movies from plot synopses. We propose a novel neural network model that merges information from synopses and emotion flows throughout the plots to predict a set of tags for movies. We compare our system with multiple baselines and found that the addition of emotion flows boosts the performance of the network by learning ≈18% more tags than a traditional machine learning system.

## 1 Introduction

User generated tags for online items are beneficial for both of the users and content providers in modern web technologies. For instance, the capability of tags in providing a quick glimpse of items can assist users to pick items precisely based on their taste and mood. On the other hand, such strength of tags enables them to act as strong search keywords and efficient features for recommendation engines (Lambiotte and Ausloos, 2006; Szomszor et al., 2007; Li et al., 2008; Borne, 2013). As a result, websites for different medias like photography[1], literature[2], film[3], and music[4] have adopted this system to make information retrieval easier. Such systems are often referred as Folksonomy (Vander Wal, 2005), social tagging, or collaborative tagging.

In movie review websites, it is very common that people assign tags to movies after watching them. Tags for movies often represent summarized characteristics of the movies such as emotional experiences, events, genre, character types, and psychological impacts. As a consequence, tags for movies became remarkably convenient for recommending movies to potential viewers based on their personal preferences and user profiles. However, this situation is not the same for all of the movies. Popular movies usually have a lot of tags as they tend to reach a higher number of users in these sites. On the other hand, low profile movies that fail to reach such an audience have very small or empty tagsets. In an investigation, we found that ≈34% of the movies among the top ≈130K movies of 22 genres[5] in IMDB do not have any tag at all. It is very likely that lack of descriptive tags negatively affects chances of movies being discovered.

An automatic process to create tags for movies by analyzing the written plot synopses or scripts could help solve this problem. Such a process would reduce the dependency on humans to accumulate tags



[1] http://www.flickr.com
[2] http://www.goodreads.com
[3] http://www.imdb.com
[4] http://www.last.fm
[5] http://www.imdb.com/genre/



*Proceedings of the 27th International Conference on Computational Linguistics*, pages 2879–2891
Santa Fe, New Mexico, USA, August 20-26, 2018.

for movies. Additionally, learning the characteristics of a movie plot and possible emotional experiences from the written synopsis is also an interesting problem by itself from the perspective of computational linguistics. As the attributes of movies are multi-dimensional, a tag prediction system for movies has to generate multiple tags for a movie. The application of predicting multiple tags from textual description is not necessarily limited to the domain of movie recommendation but also appropriate in other domains, such as video games and books, where storytelling is relevant. In this paper, we explore the problem of analyzing plot synopses to generate multiple plot-related tags for movies. Our key contributions in this paper are as follows:

- We create a neural system for predicting tags from narrative texts and provide a robust comparison against traditional machine learning systems. Table 1 shows examples of predicted tags by our system for four movies.
- We propose a neural network model that encodes flow of emotions in movie plot synopses. This emotion flow helps the model to learn more attributes of movie plots.
- We release our source code and a live demo of the tag prediction system at `http://ritual.uh.edu/folksonomication-2018`.

| IMDB ID | Movie Title | Predicted Tags |
|---|---|---|
| tt0133093 | The Matrix | though-provoking, action, sci-fi, suspenseful, mystery |
| tt0233298 | Batman Beyond: Return of the Joker | action, good versus evil, suspenseful, humor, thought-provoking |
| tt0309820 | Luther | murder, melodrama, intrigue, historical fiction, christian film |
| tt0163651 | American Pie | adult comedy, cute, feel-good, prank, entertaining |

Table 1: Example of predicted tags from the plot synopses of four movies. Blue and red labels indicate true positives and false positives respectively.

## 2 Related Work

Automatic tag generation from content-based analysis has drawn attention in different domains like music and images. For example, creating tags for music has been approached by utilizing lyrics (van Zaanen and Kanters, 2010; Hu et al., 2009), acoustic features from the tracks (Eck et al., 2008; Dieleman and Schrauwen, 2013), categorical emotion models (Kim et al., 2011), and deep neural models (Choi et al., 2017).

AutoTag (Mishne, 2006) and TagAssist (Sood et al., 2007), which utilize the text content to generate tags, aggregate information from similar blog posts to compile a list of ranked tags to present to the authors of new blog posts. Similar works (Katakis et al., 2008; Lipczak, 2008; Tatu et al., 2008) focused on recommending tags to users of BibSonomy[6] upon posting a new web page or publication as proposed systems in the ECML PKDD Discovery Challenge 2008 (Hotho et al., 2008) shared task. These systems made use of some kind of out of content resources like user metadata, and tags assigned to similar resources to generate tags.

Computational narrative studies deal with representing natural language stories by computational models that can be useful to understand, represent, and generate stories computationally. Current works attempt to model narratives using the character's personas and roles (Valls-Vargas et al., 2014; Bamman et al., 2013), interaction information between the characters (Iyyer et al., 2016; Chaturvedi et al., 2016; Chaturvedi et al., 2017) and events taking place throughout the stories (Goyal et al., 2010; Finlayson, 2012; McIntyre and Lapata, 2010). Other works try to build social networks of the characters (Agarwal et al., 2013a; Agarwal et al., 2013b; Agarwal et al., 2014; Krishnan and Eisenstein, 2015). Only a few works explored the possible type of impressions narrative texts can create on their consumers. For instance, different types of linguistic features have been used for success prediction for books (Ganjigunte Ashok et al., 2013; Maharjan et al., 2017) and tag prediction of movies from plot synopses (Kar et al., 2018). The tag prediction system predicts a fixed number of tags for each movie. But the tag space created by the system for the test data covers only 73% tags of the actual tagset as the system could capture a small portion of the multi-dimensional attributes of movie plots.

[6] `https://www.bibsonomy.org`

## 3 Dataset

We conduct our experiments on the Movie Plot Synopses with Tags (MPST) corpus (Kar et al., 2018), which is a collection of plot synopses for 14,828 movies collected from IMDb and Wikipedia. Most importantly, the corpus provides one or more fine-grained tags for each movie. The reason behind selecting this particular dataset is two-fold. First, the tagset is comprised of manually curated tags. These tags express only plot-related attributes of movies (e.g. suspenseful, violence, and melodrama) and are free of any tags foreign to the plots, such as metadata. Furthermore, grouping semantically similar tags and representing them by generalized tags helped to reduce the noise created by redundancy in tag space. Second, the corpus provides adequate amount of texts in the plot synopses as all the synopses have at least ten sentences. We follow the same split provided with the corpus, using 80% for training and 20% for test set. Table 2 gives statistics of the dataset.

| Split | #Plot Synopses | #Tags | #Tags per Movie | #Sentence per Synopsis | #Words per Synopsis |
|---|---|---|---|---|---|
| Train | 11862 | 71 | 2.97 | 42.36 | 893.39 |
| Test | 2966 | 71 | 3.04 | 42.61 | 907.96 |

Table 2: Statistics of the MPST corpus.

## 4 Encoding Emotion Flow with a Neural Network

Our proposed model simultaneously takes the emotion flow throughout the storyline and the text-based representation of the synopsis to retrieve relevant tags for a movie. Figure 1 shows the proposed architecture. The proposed neural architecture has three modules. The first module uses a convolutional neural network (CNN) to learn plot representations from synopses. The second module models the flow of emotions via a bidirectional long short-term memory (Bi-LSTM) network. And the last module contains hidden dense layers that operate on the combined representations generated by the first and second modules to predict the most likely tags for movies.

**(a) Convolutional Neural Network (CNN)**: Recent successes in different text classification problems motivated us to extract important word level features using convolutional neural networks (CNNs) (dos Santos and Gatti, 2014; Kim, 2014; Zhang et al., 2015; Kar et al., 2017; Shrestha et al., 2017). We design a model that takes word sequences as input, where each word is represented by a 300-dimensional word embedding vector. We use randomly initialized word embeddings but also experiment with the FastText[7] word embeddings trained on Wikipedia using subword information. We stack 4 sets of one-dimensional convolution modules with 1024 filters each for filter sizes 2, 3, 4, and 5 to extract word-level $n$-gram features (Kim, 2014; Zhang et al., 2015). Each filter of size $c$ is applied from window $t$ to window $t + c - 1$ on a word sequence $x_1, x_2, \ldots, x_n$. Convolution units of filter size $c$ calculate a convolution output using a weight map $W_c$, bias $b_c$, and the ReLU activation function (Nair and Hinton, 2010). The output of this operation is defined by:

$$h_{c,t} = ReLU(W_c x_{t:t+c-1} + b_c) \tag{1}$$

The ReLU activation function is defined by:

$$ReLU(x) = max(0, x) \tag{2}$$

Finally, each convolution unit produces a high-level feature map $h_c$.

$$h_c = [h_{c,1}, h_{c,2}, ..., h_{c,T-c+1}, ] \tag{3}$$

On those feature maps, we apply max-over-time pooling operation and take the maximum value as the feature produced a particular filter. We concatenate the outputs of the pooling operation for four filter sets that represent the feature representations for each plot synopsis.

**(b) CNN with Flow of Emotions (CNN-FE):** Stories can be described in terms of emotional shapes (Vonnegut, 1981), and it has been shown that the emotional arcs of stories are dominated by six

[7] https://fasttext.cc/docs/en/english-vectors.html

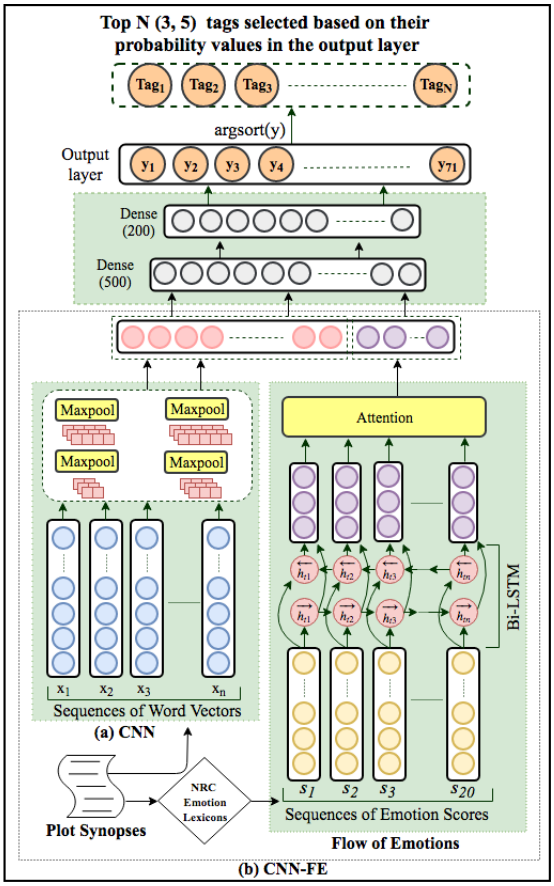


Figure 1: Convolutional Neural Network with Emotion Flow. The entire model is a combination of three modules. Module (a) learns feature representations from synopses using convolutional neural network. Module (b) incorporates emotion flows with module (a) to generate a combined representation of synopses. Module (c) uses these representations to predict the likelihood of each tag.

different shapes (Reagan et al., 2016). We believe that capturing the emotional ups and downs throughout the plots can help better understand how the story unfolds. This will enable us to predict relevant tags more accurately. So we design a neural network architecture that tries to learn representations of plots using the vector space model of words combined with the emotional ups and downs of plots.

Human emotion is a complex phenomenon to define computationally. The Hourglass of Emotions model (Cambria et al., 2012) categorized human emotions into four affective dimensions (*attention, sensitivity, aptitude, and pleasantness*), which started from the study of human emotions by Plutchik (2001). Each of these affective dimensions is represented by six different activation levels that make up to 24 distinct labels called 'elementary emotions' that represent the total emotional state of the human mind. NRC[8] emotion lexicons (Mohammad and Turney, 2013) is a list of 14,182 words[9] and their binary associations with eight types of elementary emotions from the Hourglass of Emotions model (*anger, anticipation, joy, trust, disgust, sadness, surprise,* and *fear*) with polarity. These lexicons have been used effectively in tracking the emotions in literary texts (Mohammad, 2011) and predicting success of books (Maharjan et al., 2018).

To model the flow of emotions throughout the plots, we divide each synopsis into $N$ equally-sized segments based on words. For each segment, we compute the percentage of words corresponding to each emotion and polarity type (positive and negative) using the NRC emotion lexicons. More precisely, for a synopsis $x \epsilon X$, where $X$ denotes the entire collection of plot synopses, we create $N$ sequences of emotion vectors using the NRC emotion lexicons as shown below:

$$x \rightarrow s_{1:N} = [s_1, s_2, ..., s_N] \tag{4}$$

[8]National Research Council Canada

[9]Version 0.92

where $s_i$ is the emotion vector for segment $i$. We experiment with different values of $N$, and $N = 20$ works better on the validation data.

As recurrent neural networks are good at encoding sequential data, we feed the sequence of emotion vectors into a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with 16 units as shown in Figure 1. This bidirectional LSTM layer tries to summarize the contextual flow of emotions from both directions of the plots. The forward LSTMs read the sequence from $s_1$ to $s_N$, while the backward LSTMs read the sequence in reverse from $s_N$ to $s_1$. These operations will compute the forward hidden states $(\overrightarrow{h_1}, \ldots, \overrightarrow{h_N})$ and backward hidden states $(\overleftarrow{h_1}, \ldots, \overleftarrow{h_N})$. For input sequence $s$, the hidden states $h_t$ are computed using the following intermediate calculations:

$$
\begin{aligned}
i_t &= \sigma(W_{si}s_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{sf}s_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(W_{sc}s_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{sc}s_t + W_{hc}h_{t-1} + b_c) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

where, $W$ and $b$ denote the weight matrices and bias, respectively. $\sigma$ is the sigmoid activation function, and $i$, $f$, $o$, and $c$ are *input gate, forget gate, output gate*, and *cell* activation vectors, respectively. The annotation for each segment $s_i$ is obtained by concatenating its forward hidden states $\overrightarrow{h_i}$ and backward hidden states $\overleftarrow{h_i}$, i.e. $h_i$=$[\overrightarrow{h_i}; \overleftarrow{h_i}]$. We then apply attention mechanism on this representation to get a unified representation of the emotion flow.

Attention models have been used effectively in many problems related to computer vision (Mnih et al., 2014; Ba et al., 2014) and have been successfully adopted in problems related to natural language processing (Bahdanau et al., 2014; Seo et al., 2016). An attention layer applied on top of a feature map $h_i$ computes the weighted sum $r$ as follows:

$$r = \sum_i \alpha_i h_i \tag{5}$$

and the weight $\alpha_i$ is defined as

$$\alpha_i = \frac{\exp(score(h_i))}{\sum_{i'} \exp(score(h_{i'}))}, \tag{6}$$

where, $score(.)$ is computed as follows:

$$score(h_i) = v^T tanh(W_a h_i + b_a) \tag{7}$$

where, $W$, $b$, $v$, and $u$ are model parameters. Finally, we concatenate the representation of the emotion flow produced by the attention operation and the output vector with the vector representation generated from the CNN module.

The concatenated vector is then fed into two hidden dense layers with 500 and 200 neurons. To improve generalization of the model, we use dropout with a rate of 0.4 after each hidden layer. Finally, we add the output layer $\hat{y}$ with 71 neurons to compute predictions for 71 tags. To overcome the imbalance of the tags, we weight the posterior probabilities for each tag using different weight values. Weight value $CW_t$ for tag $t \epsilon T$ is defined by,

$$CW_t = \frac{|D|}{|T| \times M_t} \tag{8}$$

where, $|D|$ is the size of the training set, $|T|$ is the number of classes, and $M_t$ is the number of movies having tag $t$ in the training set. We normalize the output layer by applying a softmax function defined by,

$$softmax(\hat{y}) = \frac{exp(\hat{y})}{\sum_{k=0}^{70} exp(\hat{y}_k)} \tag{9}$$

Based on the ranking for each tag, we then select top $N$ (3/5/10) tags for a movie.

## 5 Experimental Setup

**Data Processing and Training:** As a preprocessing step, we lowercase the synopses, remove stop-words and also limit the vocabulary to top 5K words to reduce noise and data sparsity. Then we convert each synopsis into a sequence of 1500 integers where each integer represents the index of the corresponding word in the vocabulary. For the sequences longer than 1500 words, we truncate them from the left based on experiments on the development set. Shorter sequences are left padded with zeros.
During training, we use 20% of the training data as validation data. We tune various deep model parameters (dropouts, learning rate, weight initialization schemes, and batch size) using early stopping technique on the validation data. We use the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) to compute the loss between the true and predicted tag distributions and train the network using the RMSprop optimization algorithm (Tieleman and Hinton, 2012) with a learning rate of 0.0001. We implemented our neural network using the PyTorch deep learning framework[10].
**Baselines:** We compare the model performance against three baselines: majority baseline, random baseline, and traditional machine learning system. The majority baseline method assigns the most frequent three or five or ten tags in the training set to all the movies. Similarly, the random baseline assigns randomly selected three or five or ten tags to each movie. Finally, we compare our results with the benchmark system reported in Kar et al. (2018). This benchmark system used different types of hand-crafted lexical, semantic, and sentiment features to train a OneVsRest approach model with logistic regression as the base classifier.
**Evaluation Measures:** We try to follow the same evaluation methodology as described in Kar et al. (2018). We create two sets of tags for each movie by choosing the most likely three and five tags by the system. Additionally, we report our results on a wider range of tags, where we select top ten predictions. We evaluate the performance using the number of unique tags learned by the system (TL), micro averaged F1, and tag recall (TL). Tags learned (TL) computes how many unique tags are being predicted by the system for the test data (size of the tag space created by the model for test data). Tag recall represents the average recall per tag and it is defined by the following equation:

$$TR = \frac{\sum_{i=1}^{|T|} |R_i|}{|T|} \tag{10}$$

Here, $|T|$ is the total number of tags in the corpus, and $R_i$ is the recall for the $i^{th}$ tag.

## 6 Results and Discussions

| | Top 3 | | | Top 5 | | | Top 10 | | |
|---|---|---|---|---|---|---|---|---|---|
| **Methods** | **TL** | **F1** | **TR** | **TL** | **F1** | **TR** | **TL** | **F1** | **TR** |
| Baseline: Most Frequent | 3 | 29.7 | 4.23 | 5 | 28.4 | 14.08 | 10 | 28.4 | 13.73 |
| Baseline: Random | 71 | 4.2 | 4.21 | 71 | 6.4 | 15.04 | 71 | 6.6 | 14.36 |
| Baseline: Kar et al. (2018) | 47 | **37.3** | **10.52** | 52 | **37.3** | **16.77** | — | — | — |
| CNN without class weights | 24 | 36.8 | 7.99 | 26 | 36.7 | 12.62 | 27 | **31.3** | 24.52 |
| CNN with class weights | 49 | 34.9 | 9.85 | 55 | 35.7 | 14.94 | 67 | 30.8 | **26.86** |
| CNN-FE | **58** | 36.9 | 9.40 | **65** | 36.7 | 14.11 | **70** | 31.1 | 24.76 |
| CNN-FE + FastText | 53 | **37.3** | 10.00 | 59 | 36.8 | 15.47 | 63 | 30.6 | 26.45 |

Table 3: Performance of tag prediction systems on the test data. We report results of two setups using three matrices (TL: Tags learned, F1: Micro f1, TR: Tag recall).

Table 3 shows our results for Top 3, Top 5, and Top 10 settings. We will mainly discuss the results achieved by selecting top five tags as it allows us to compare with all the baseline systems and more tags to discuss about. As the most frequent baseline system assigns a fixed set of tags to all the movies, it fails to exhibit diversity in the created tag space. Still it manages to achieve a micro-F1 score around 28%. On the other hand, the random baseline system creates the most diverse tag space by using all of the possible tags. However its lower micro-F1 score of 6.30% makes it impractical to be used in real world scenario.

[10] https://pytorch.org

At this point, we find an interesting trade-off between accuracy and diversity. It is expected that a good movie tagger will be able to capture the multi-dimensional attributes of the plots that allows to generalize a diverse tag space. Tagging a large collection of movies with a very small and fixed set of tags (e.g. majority baseline system) is not useful for either a recommendation system or users. Equally important is the relevance between the movies and the tags created for those movies. The hand-crafted features based approach (Kar et al., 2018) achieves a micro-F1 around 37%, which outperforms the majority and random baselines. But the system was able to learn only 52 tags, which makes 73% of the total tags.

Our approach achieves a lower micro-F1 score than the traditional machine learning one, but it performs better in terms of learning more tags. We observe that the micro-F1 of the CNN model with only word sequences is very close (36.7%) to the hand-crafted features based system. However, it is able to learn only around 37% of the tags. By utilizing class weights in this model (see Eq. 8), we improve the learning for under-represented tags yielding an increase in *tag recall* (TR) and *tags learned* (TL). But the micro-f1 drops to 35.7%. With the addition of emotion flows to CNN, the CNN-FE model learns significantly more tags while micro-F1 and tag recall do not change much. Initializing the embedding layer with pre-trained embeddings made a small improvement in micro-F1 but the model learns comparatively lesser tags. If we compare the CNN-FE model with the hand-crafted feature based system, micro-F1 using CNN-FE is slightly lower ($\approx 1\%$) than the feature based system. But it provides a strong improvement in terms of the number of tags it learns (TL). CNN-FE learns around 91% tags of the tagset compared to 73% with the feature based system. It is an interesting improvement, because model is learning more tags and it is better at assigning relevant tags to movies. We observe similar pattern for the rest of the two sets of tags where we select top three and ten tags. For all the sets, CNN-FE model learns the highest number of tags compared to the other models. In terms of micro-F1 and tag recall, it does not achieve the highest numbers but performs very closely.

**Incompleteness in Tag Spaces**: One of the limitations of folksonomies is the incompleteness in tag spaces. The fact that users have not tagged an item with a specific label does not imply that that label does not apply to the item. Incompleteness makes learning challenging for computational models as the training and evaluation process penalizes the model for predicting a tag that is not present in the ground truth tags, even though in some cases it may be a suitable tag. For example, ground truth tags for the movie *Luther (2003)*[11] are *murder, romantic,* and *violence* (Table 1). And the predicted tags from our proposed model are *murder, melodrama, intrigue, historical fiction,* and *christian film*. The film is indeed a Christian film[12] portraying the biography of Martin Luther, who led the Christian reformation during the 16th century. According to the Wikipedia, "*Luther is a 2003 American-German epic historical drama film loosely based on the life of Martin Luther*"[13]. Similarly, Edtv[14] (Table 6) has tags *romantic* and *satire* in the dataset. Our system predicted *adult comedy* and this tag is appropriate for this movie.

In these two cases, the system will get lower micro-F1 since the relevant tags are not part of the ground truth. Perhaps a different evaluation scheme could be better suited for this task. We plan to work on this issue in our future work.

**Significance of the Flow of Emotions**: The results suggest that incorporating the flow of emotions helps to achieve better results by learning more tags. Figure 2 shows some tags with significant improvements in recall after incorporating the flow of emotions. We notice such improvements for around 30 tags. We argue that for these tags (e.g. *absurd, cruelty, thought-provoking, claustrophobic*) the changes in specific sentiments are adding new information helpful for identifying relevant tags. But we also notice negative changes in recall for around 10 tags, which are mostly related to the theme of the story (e.g. *blaxploitation, alternate history, historical fiction, sci-fi*). It will be an interesting direction of future work to add a mechanism that can also learn to discern when emotion flow should contribute more to the prediction task.

In Figure 3, we inspect how the flow of emotions looks like in different types of plots. Emotions like *joy* and *trust* are continuously dominant over *disgust* and *anger* in the plot of *Arthur (1981)*, which is

[11] http://www.imdb.com/title/tt0309820

[12] https://www.christianfilmdatabase.com/review/luther-2

[13] https://en.wikipedia.org/wiki/Luther_(2003_film)

[14] http://www.imdb.com/title/tt0131369/

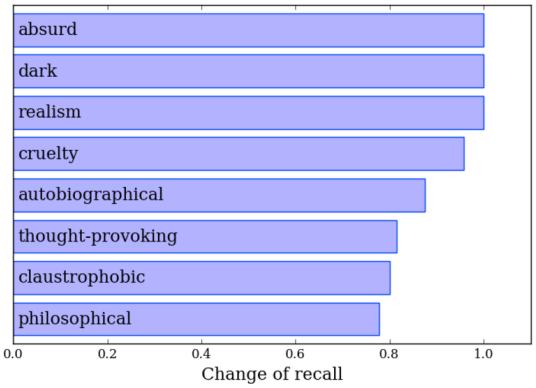


Figure 2: Tags with higher change of recall after adding the flow of emotions in CNN.

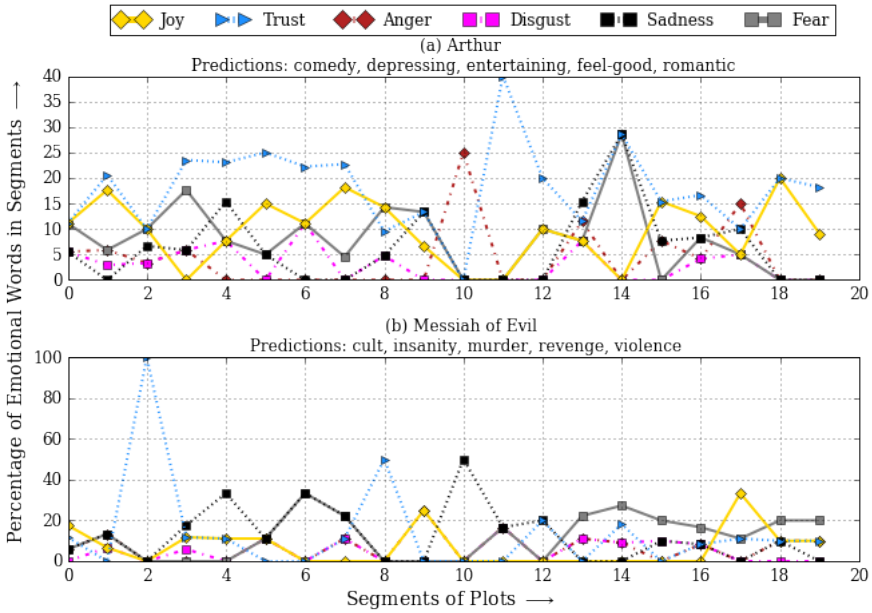


Figure 3: The flow of emotions in the plots of 2 different types of movies. Each synopsis was divided into 20 segments based on the words, and percentage of the emotions for each segment was calculated using the NRC emotion lexicons. The y axis represents the percentage of emotions in each segment; whereas, the x axis represents the segments.

a comedy film. We can observe sudden spikes in *sadness* and *fear* at segment 14, which is the possible reason for triggering the tag *depressing*. We observe a different pattern in the flow of emotions in *Messiah of Evil (1973)*, which is a horror film. Here the dominant emotions are *sadness* and *fear*. Such characteristics of emotions are helpful to determine the type and possible experiences from a movie. Our model seems to be able to leverage this information that is allowing it to learn more tags; specifically tags that are related to feelings.

**Learning or Copying?** We found that only 11.8% of the 14,830 predicted tags for the ∼3K movies in the test data were found in the synopses themselves. 12.7% of the total 9,022 ground truth tags appear in the plot synopses. These numbers suggest that the model is not dependent on the occurrences of the tags in the synopses to make predictions, rather it seems it is trying to understand the plots and assign tags based on that. We also found that all the tags that were present in the synopses of the test data are also present in the synopses of the training data. Then we investigate what type of tags appear in the synopses and which ones do not. Tags present in the synopses are mostly genre or event related tags like *horror, violence, historical*. On the other hand, most of the tags that do not appear in the synopses are the tags that require a more sophisticated analysis of the plots synopses (e.g. *thought-provoking, feel-good,*

*suspenseful*). It is not necessarily bad to predict tags that are in the synopses, since they are still useful for recommender systems. However, if this was the only ability of the proposed models, their value would be limited. Luckily this analysis, and the results presented earlier show that the model is able to infer relevant tags, even if they have not been observed in the synopses. This is a much more interesting finding.

**Learning Stories from Different Representations:** Movie scripts represent the detailed story of a movie, whereas the plot synopses are summaries of the movie. The problem with movie scripts is that they are not as readily available as plot synopses. However, it is still interesting to evaluate our approach to predict tags from movie scripts. For this purpose, we collected movie scripts from our test set. We

| | Top 3 | | | Top 5 | | |
|---|---|---|---|---|---|---|
| | **F1** | **TR** | **TL** | **F1** | **TR** | **TL** |
| Plot Synopses | 29.3 | 8.04 | 28 | 38.7 | 15.70 | 35 |
| Scripts | 29.8 | 5.16 | 19 | 37.0 | 9.27 | 26 |

Table 4: Evaluation of predictions using plot synopses and scripts

were able to find 80 movie scripts using the ScriptBase corpus (Gorinski and Lapata, 2015).

In table 4, we show the evaluation of tags generated using plot synopses and scripts. Despite having similar micro-f1 scores, *tag recall* and *tags learned* are lower when we use the scripts. A possible explanation for this is the train/test mismatch since the model was trained using summarized versions of the movie, while the test data contained full movies scripts. Additional sources of error could come from the external info included in scripts (such as descriptions of actions from the characters or settings).

| **Percentage of Match** | **Percentage of Movies** |
|---|---|
| >=80% | 40% |
| >=40% & <80% | 47.5% |
| >=20% & <40% | 11.25% |

Table 5: Percentage of the match between the sets of top five tags generated from the scripts and plot synopses.

Table 5 shows that for most of the movies we generate very similar tags using the scripts and plot synopses. For 40% movies, at least 80% tags are the same. While the predictions are not identical, these results show a consistency in the learned tags from our system. An interesting direction for future work would be to study what aspects in a full movie script are relevant to predict tags.

| |
|---|
| **Title:** A Nightmare on Elm Street 5: The Dream Child<br>**Ground Truths:** cult, good versus evil, insanity, murder, sadist, violence<br>**Synopsis:** cult, murder, paranormal, revenge, violence<br>**Script:** murder, violence, flashback, cult, suspenseful |
| **Title:** EDtv<br>**Ground Truths:** romantic, satire<br>**Synopsis:** adult comedy, comedy, entertaining, prank, satire<br>**Script:** comedy, satire, prank, entertaining, adult comedy |
| **Title:** Toy Story<br>**Ground Truths:** clever, comedy, cult, cute, entertaining, fantasy, humor, violence<br>**Synopsis:** comedy, cult, entertaining, humor, psychedelic<br>**Script:** psychedelic, comedy, entertaining, cult, absurd |
| **Title:** Margot at the Wedding<br>**Ground Truths:** romantic, storytelling, violence<br>**Synopsis:** depressing, dramatic, melodrama, queer, romantic<br>**Script:** psychological, murder, mystery, flashback, insanity |

Table 6: Example of ground truth tags of movies from the test set and the generated tags for them using plot synopses and scripts.

**Challenging Tags:** We found that these seven tags: *stupid, grindhouse film, blaxploitation, magical realism, brainwashing, plot twist,* and *allegory*, were not assigned to any movies in the test set. One reason might be that these are very infrequent (around 0.06% of movies have them assigned as their tags). This

will obviously make them difficult to learn. Again, these are subjective as well. We believe that tagging a plot as stupid or brainwashing is complicated and depends on perspectives of a tagger. We plan to investigate such type of tags in the future.

## 7 Conclusions and Future Work

In this paper we explore the problem of automatically creating tags for movies using plot synopses. We propose a model that learns word level feature representations from the synopses using CNNs and models sentiment flow throughout the plots using a bidirectional LSTM. We evaluated our model on a corpus that contains plot synopses and tags of 14K movies. We compared our model against a majority and random baselines, and a system that uses traditional hand-crafted linguistic features. We found that incorporating emotion flows boosts prediction performance by improving the learning of tags related to feelings as well as increasing the overall number of tags learned.
Predicting tags for movies is an interesting and complicated problem at the same time. To further improve our results, we plan to investigate more sophisticated architectures and explore ways to tackle the problem of incompleteness in the tag space. We also plan to evaluate the quality of predicted tags using a human study evaluation and experiment on predicting tags in other storytelling related domains.

## Acknowledgements

This work was partially supported by the National Science Foundation under grant number 1462141 and by the U.S. Department of Defense under grant W911NF-16-1-0422.

## References

Apoorv Agarwal, Anup Kotalwar, and Owen Rambow. 2013a. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1202–1208, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Apoorv Agarwal, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. 2013b. Sinnet: Social interaction network extractor from text. In *The Companion Volume of the Proceedings of IJCNLP 2013: System Demonstrations*, pages 33–36, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Apoorv Agarwal, Sriramkumar Balasubramanian, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. 2014. Frame semantic tree kernels for social network extraction from text. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 211–219, Gothenburg, Sweden, April. Association for Computational Linguistics.

Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2014. Multiple object recognition with visual attention. *CoRR*, abs/1412.7755.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria, August. Association for Computational Linguistics.

Kirk Borne. 2013. Collaborative annotation for scientific data discovery and reuse. *Bulletin of the American Society for Information Science and Technology*, 39(4):44–45.

Erik Cambria, Andrew Livingstone, and Amir Hussain. 2012. The hourglass of emotions. In *Proceedings of the 2011 International Conference on Cognitive Behavioural Systems*, COST'11, pages 144–157, Berlin, Heidelberg. Springer-Verlag.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daumé III, and Chris Dyer. 2016. Modeling evolving relationships between characters in literary novels. In Dale Schuurmans and Michael P. Wellman, editors, *AAAI*, pages 2704–2710. AAAI Press.

Snigdha Chaturvedi, Mohit Iyyer, and Hal Daumé III. 2017. Unsupervised learning of evolving relationships between literary characters.

K. Choi, G. Fazekas, M. Sandler, and K. Cho. 2017. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396, March.

Sander Dieleman and Benjamin Schrauwen. 2013. Multiscale approaches to music audio feature learning. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, November 4-8. http://www.ppgia.pucpr.br/ismir2013/wp-content/uploads/2013/09/69_Paper.pdf.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Douglas Eck, Paul Lamere, Thierry Bertin-mahieux, and Stephen Green. 2008. Automatic generation of social tags for music recommendation. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 385–392. Curran Associates, Inc.

Mark Alan Finlayson. 2012. *Learning narrative structure from annotated folktales*. Ph.D. thesis, Massachusetts Institute of Technology.

Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with style: Using writing style to predict the success of novels. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1764, Seattle, Washington, USA, October. Association for Computational Linguistics.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado, May–June. Association for Computational Linguistics.

Amit Goyal, Ellen Riloff, and Hal Daumé, III. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 77–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Andreas Hotho, Dominik Benz, Robert Jáschke, and Beate Krause. 2008. Ecml pkdd discovery challenge 2008 (rsdc'08). In *Workshop at 18th Europ. Conf. on Machine Learning (ECML'08)/11th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'08)*, volume 32.

Xiao Hu, J. Stephen Downie, and Andreas F. Ehmann. 2009. Lyric text mining in music mood classification. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009*, pages 411–416.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544. Association for Computational Linguistics.

Sudipta Kar, Suraj Maharjan, and Thamar Solorio. 2017. RiTUAL-UH at semeval-2017 task 5: Sentiment analysis on financial data using neural networks. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 877–882.

Sudipta Kar, Suraj Maharjan, A. Pastor López-Monroy, and Thamar Solorio. 2018. MPST: A corpus of movie plot synopses with tags. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, May. European Language Resources Association (ELRA).

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD 2008 Discovery Challenge*.

JungHyun Kim, Seungjae Lee, SungMin Kim, and Won Young Yoo. 2011. Music mood classification model based on arousal-valence values. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pages 292–295. IEEE.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Vinodh Krishnan and Jacob Eisenstein. 2015. "You're Mr. Lebowski, I'm the Dude": Inducing address term formality in signed social networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1616–1626, Denver, Colorado, May–June. Association for Computational Linguistics.

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03.

Renaud Lambiotte and Marcel Ausloos, 2006. *Collaborative Tagging as a Tripartite Network*, pages 1114–1117. Springer Berlin Heidelberg, Berlin, Heidelberg.

Xin Li, Lei Guo, and Yihong Eric Zhao. 2008. Tag-based social interest discovery. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 675–684, New York, NY, USA. ACM.

Marek Lipczak. 2008. Tag recommendation for folksonomies oriented towards individual users. In *In: Proc. of the ECML PKDD Discovery Challenge*.

Suraj Maharjan, John Arevalo, Manuel Montes, Fabio A González, and Thamar Solorio. 2017. A multi-task approach to predict likability of books. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1217–1227.

Suraj Maharjan, Sudipta Kar, Manuel Montes, Fabio A. Gonzalez, and Thamar Solorio. 2018. Letting emotions flow: Success prediction by modeling the flow of emotions in books. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 259–265, New Orleans, Louisiana, June. Association for Computational Linguistics.

Neil McIntyre and Mirella Lapata. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1562–1572, Uppsala, Sweden, July. Association for Computational Linguistics.

Gilad Mishne. 2006. Autotag: A collaborative approach to automated tag assignment for weblog posts. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 953–954, New York, NY, USA. ACM.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. *CoRR*, abs/1406.6247.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

Saif Mohammad. 2011. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH '11, pages 105–114, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA. Omnipress.

Robert Plutchik. 2001. The nature of emotions human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4):344–350.

Andrew J. Reagan, Lewis Mitchell, Dilan Kiley, Christopher M. Danforth, and Peter Sheridan Dodds. 2016. The emotional arcs of stories are dominated by six basic shapes. *CoRR*, abs/1606.07772.

Paul Hongsuck Seo, Zhe Lin, Scott Cohen, Xiaohui Shen, and Bohyung Han. 2016. Hierarchical attention networks. *CoRR*, abs/1606.02393.

Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674, Valencia, Spain. Association for Computational Linguistics.

Sanjay C. Sood, Kristian J. Hammond, Sara H. Owsley, and Larry Birnbaum, 2007. *TagAssist: Automatic tag suggestion for blog posts*.

Martin Szomszor, Ciro Cattuto, Harith Alani, Kieron O'Hara, Andrea Baldassarri, Vittorio Loreto, and Vito D.P. Servedio. 2007. Folksonomies, the semantic web, and movie recommendation.

M. Tatu, M. Srikanth, and T. D'Silva. 2008. *RSDC'08: Tag Recommendations using Bookmark Content*. Workshop at 18th Europ. Conf. on Machine Learning (ECML'08) / 11th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'08).

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontanón. 2014. Toward automatic role identification in unannotated folk tales. In *Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 188–194. AAAI Press.

Menno van Zaanen and Pieter Kanters. 2010. Automatic mood classification using tf*idf based on lyrics. In *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, August 9-13, 2010*, pages 75–80.

Thomas Vander Wal. 2005. Folksonomy definition and wikipedia. *vanderwal. net*.

Kurt Vonnegut. 1981. Palm sunday: An autobiographical collage.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.