Active Learning for Data-efficient Training on Sentiment Classification

Stanford CS224N Custom Project

Kasha Akrami
Department of Computer Science
Stanford University
kakrami@stanford.edu

Kevin Wang
Department of Computer Science
Stanford University
kevwang1@stanford.edu

Amber Yang
Department of Computer Science
Stanford University
yanga@stanford.edu

Abstract

Our project is motivated by exploring the effectiveness of active learning techniques in order to improve the speed of learning for sentiment classification. We specifically adapted a bag-of-words classifier with the goal of labeling IMDB movie reviews according to their sentiment. To perform classification, our neural network used a simple hidden layer. We compared three active learning techniques — uncertainty-based, Bayesian active learning, and data distance. Compared to the non-active learning baseline, the most effective active learning techniques demonstrated significant improvement in accuracy.

1 Key Information to include

- Mentor: Kathy Yu
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

Currently with most NLP tasks, there are often massive quantities of unlabelled data that take time and cost to label. In some domains, such as medical text, it is highly costly to label data, preventing wide scale adoption of NLP applications in the domain. Consequently, the chance to reduce the amount of data needed to be labeled is a valuable area of research within NLP as it can help speed up NLP tasks. Because this is an active area of research within NLP, and since there remains outstanding consensus on what is the most effective way to label data efficiently, we thought this would be a worthwhile problem to explore as part of our final project for CS224n.

Active learning is a model training procedure that alternates between model training and data labeling/acquisition. In the most common approach, pool-based sampling, instances are drawn from the unlabelled pool by some ranking procedure. After training each subsequent portion of data, more data is chosen from the pool until the labelling budget runs out. Another active learning approach is membership query synthesis, in which the learner generates instances to be labelled. For example, the learner may generate new text samples that it is uncertain about, which are then labelled. Furthermore, another approach, stream-based selective sampling, gives the learner one sample at a time, and the learner chooses whether to label or reject the sample.

We took the approach of using pool-based active learning, with inspiration from the the paper "Active Learning for Visual Question Answering: An Empirical Study." We incorporated three different ranking procedures for unlabelled data, adapting these techniques to work in our NLP sentiment classification scenario. Because active learning is a general approach that often operates data and model-agnostically, these approaches transferred well to the NLP scenario.

3 Related Work

There are several related research paper that explore the usage of active learning for both deep learning and NLP tasks. One method proposed by Patrick Hemmer, Niklas Kuhl, and Jakob Schoffer in the research paper "DEAL: Deep Evidential Active Learning for Image Classification" is to capture high prediction uncertainty in order to efficiently learn from unlabeled, which they called DEAL.[2] In DEAL, the underlying softmax standard output distribution of the Convolutional Neural Network (CNN) with a Dirichlet density distribution. This allowed data points that contributed to more efficient learning to be easier identified. DEAL was deved on both synthetic and real datasets and was shown to outperform other state-of-the art active learning methods. For example, when DEAL was compared to the active learning method minimal margin with softmax in identifying pneumonia in a set of lung images, it performed 1.76% better.

Another method proposed in the research paper "Learning Active Learning from Data" by Ksenia Konyushkova, Sznitman Raphael, and Pascal Fua is to train a regressor that predicts the expected error reduction for a candidate sample in a particular learning state[3]. Our approach looked at three active-learning tasks: cramming, curiosity-driven learning, and goal-driven learning. Konyushkova Et al. took a markedly different approach: by taking a trained classifier and its output for a specific sample without a label, they were able to predict the reduction in generalization error that can be expected by adding the label to that datapoint. They called this approach "Learning Active Learning" (LAL). LAL, since it was formed as a regression problem, was able to perform better classification than other active learning strategies like random sampling and uncertainty sampling.

With regards to improving current active learning strategies rather than creating a new one, the paper "Class-Balanced Active Learning for Image Classification" by Javad Zolfaghari Bengar, Joost van de Weijer, Laura Lopez Fuentes, and Bogdan Raducanu provided a sound method for improvement[1]. They proposed a a general optimization framework that took "class-balancing" (p. 1) into account. While current active learning methods will process unlabeled data into smaller batches using cycles with no order, their method of class-imbalance used a batch mode variant. Active learning methods such as entropy, k-center-greedy, variational adversarial active learning that used class-balancing showed greater accuracy during image classification tasks.

4 Approach

The goal of our research is to compare the accuracy results of three different information-theoretic active learning strategies: cramming (uncertainty-based), curiosity-driven learning (Bayesian active learning), and goal-driven (data distance) learning applied to the task of classifying IMDB movie reviews as having either positive or negative sentiment. The three active learning technique results are then compared to the results of a baseline binary classification model with data obtained from random sampling to examine the effectiveness of active learning.

We will begin by detailing our active learning approaches that are slight variations of methods originally presented in our chosen research paper.

We began by pooling 50,000 IMDB movie reviews that had a label of either "positive" or "negative" sentiment. After pre-processing the data, a bag of words encoder was applied to the entire corpus of movie reviews to produce a feature vector. We then applied a single-hidden layer neural network with 1,000 features, where every unique word in the bag of words is its own index in the bag. More specifically, only the top 1000 most-frequently appearing words in the corpus of movie reviews are kept in the bag of words. The output of the neural network is a classification label of either 0

("negative" sentiment) or 1 ("positive" sentiment).

In all three active learning approaches and the baseline, we allow 5,000 training samples to be used (from the total 37500 training sample). Therefore, the quantity of data trained on is the same between all models.

4.1 Baseline Model

The non-active learning baseline model trains on the dataset as normal. The training data had 37,500 items, and a random 5000 samples were taken. The model was trained for 10 epochs and the final test accuracy was calculated.

4.2 Cramming (Uncertainty) Active Learning Technique

The cramming or "uncertainty sampling" active learning technique minimizes uncertainty (entropy) of outputs (sentiment predictions) for movie reviews in the training dataset. It selects the training samples whose output predictions' distributions have maximum entropy to train on. In other words, the model chooses the examples it is least certain about (i.e. with confidence values closest to 0.5 in the binary classification scenario) for labelling.

$$s_{entropy}(review) = -\sum_{a} P(A = a|review) log P(A = a|review) \tag{1}$$

4.3 Curiosity-driven (BALD) Active Learning Technique

The curiosity-driven or Bayesian Active Learning by Disagreement (BALD) technique randomizes the model each time it tries to choose new training data samples, and the model chooses the data sample whose output would expectedly bring the steepest decrease in model parameter uncertainty if added to the training set.

To be more precise, when the model attempts to choose which data samples it wants to label, dropout is randomly applied to the output layer of the model to measure how much change there is to the output score. If there is a large change to the output score, this is an indication that the model is not confident in that specific data sample, so the model will choose that sample to label.
In the BALD scenario, we apply 100 iterations of forward passes with dropout to calculate the distribution of change in output scores. After hyperparameter tuning, we found this number to give the best score.

4.4 Goal-driven (Data Distance) Active Learning Technique

The goal-driven or data distance active learning technique chose sample reviews in the training set that have the most number of novel words to the model. The model determines whether the words in a review are novel by computing the distance between a the bag-of-words vector for a new data item and the averaged vector for data items the model has already trained on.

The pseudocode for any of the active learning approaches is as follows. The only difference is the way $s(Q, I)$ (the ranking measure) is calculated:

Algorithm 1 Active learning for Visual Question Answering

Initialize $\mathcal{D}_{train}$ with N inital training examples. Use the rest of (Q, I) in VQA TRAIN set as pool.Q
Train $\boldsymbol{\theta}$ on $\mathcal{D}_{train}$ for K epochs using Eq. 2 for initial $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$.
for $iter = 1, \ldots, L$ **do**
    Sample $\boldsymbol{\omega} \sim q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ M times.
    Using each $\boldsymbol{\omega}$ to make predictions $P(A|Q, I, \boldsymbol{\omega})$ on all pool and test question-image pairs.
    Compute $s(Q, I)$ for every (Q, I) in pool using Eq. 4, 5 or 9.
    Select the top G high-scoring (Q, I) pairs from the pool.[2]
    Lookup answers A for (Q, I) pairs in the VQA training set (proxy for querying a human).
    Add (Q, I, A) tuples to $\mathcal{D}_{train}$.
    Update $\boldsymbol{\theta}$ on new $\mathcal{D}_{train}$ for K epochs.
end for

For training with the active learning approaches, we had five iterations of taking the 1000 highest priority samples, for a total of 5000 labelled examples. We labeled these samples then trained on them iteratively.

All the code written in our approach was original, including both the baseline and active learning models. For the active learning model however, our code was informed by the psuedocode algorithm provided above. There is an important difference between our code and the provided algorithm, which is our usage of IMDB movie reviews instead of the VQA training data.

The only part of our code that requires acknowledgment is our design of the three active learning techniques, which are informed by the pseudocode algorithms provided in our chosen reference paper.

5 Experiments

5.1 Data

Our dataset is the IMDB reviews dataset[4], which is a collection of 50,000 movie reviews scraped from the IMDB website. Each movie review is attached with a label of 'positive' (with a value of 1) or 'negative' (with a value of 0) according to the sentiment of the review. We pre-processed the dataset as described below. The data was split 80/20 between training and test sets, for a total of 37,500 items in the training set and 12,500 items in the test set. The dataset has close to a 50/50 split of positive and negative sentiment labels, so accuracy is a clear way to evaluate the effectiveness of a classifier.

5.1.1 Data preprocessing

We first cleaned each movie review to remove quotation marks and stop words. We also applied a regex pattern to standardize capitalization of words in each review such that no capital letters appear in the reviews. Then, we pre-processed the sentiment labels attached to each movie review. Instead of having a string label ('positive' or 'negative'), we converted this to a numerical label (0 corresponding to 'negative' and 1 corresponding to 'positive').

Next, we created a bag of words model with 1000 features such that each unique word in the corpus has its own unique index in the bag of words and the top 1000 words according to their frequency count in the corpus are selected. Each document now becomes a 1000 element vector.

The dataset is split into a training set that has 37500 movie reviews and a test set that has 12500 movie reviews.

5.2 Evaluation method

The loss function used in our model is implemented via the help of `torch.nn.BCEWithLogitsLoss`, and we calculated the loss every 1000 epochs. `BCEWithLogitsLoss` is a loss function commonly used for classification tasks. Specifically, the loss function can be described as:

$$l(x, y) = L = \{l_1, ..., l_N\}^T, l_n = -w_n[y_n log(x_n) + (1 - y_n) log(1 - (x_n))],$$

where N is the batch size, which is 100 in our case. We analyze the loss function for the non-active learning baseline and the three active-learning models and compared them for our analyses. These details can be found in the Results section. Furthermore, we compared the training and testing accuracy for the non-active learning and active-learning models to each other to understand the effects of active learning on model accuracy.

5.3 Experimental details

We had four different models that we implemented: random sampling strategy (which acted as our baseline), BALD, data distance, and uncertainty.

Our random sampling configuration was the most intuitive to implement. While our main implementation specifications can be found above in the Approach section, there were some specific experimental details we fine-tuned. Random sampling ran for 10 epochs.

BALD, data distance, and uncertainty were able to follow similar configurations. All three models used five active learning (in other words, they collected new data five times). In each step, they collected 1000 pieces of data from the larger pool. Furthermore, since each model ran for 10 epochs (like random sampling) but for five learning steps, this meant there were 50 total epochs that these three models ran for.

In all four of our models, we used the Adam Optimizer, the standard Pytorch gradient for the loss on the tensors (computed using the .backward() function), and a batch size of 100.

The training time for each of the four models was as follows:

Table 1: Run Times (in minutes : seconds)

Random Sampling	BALD	Data Distance	Uncertainty
0:02.718399	07:54.882449	0:14.483637	0:14.299948

This difference shows that random sampling has the fastest training time, while the active learning approaches have a longer training time. In a sense, active learning trades off reduced data labelling expense for extra computation expense (to evaluate the usefulness of a new data item).

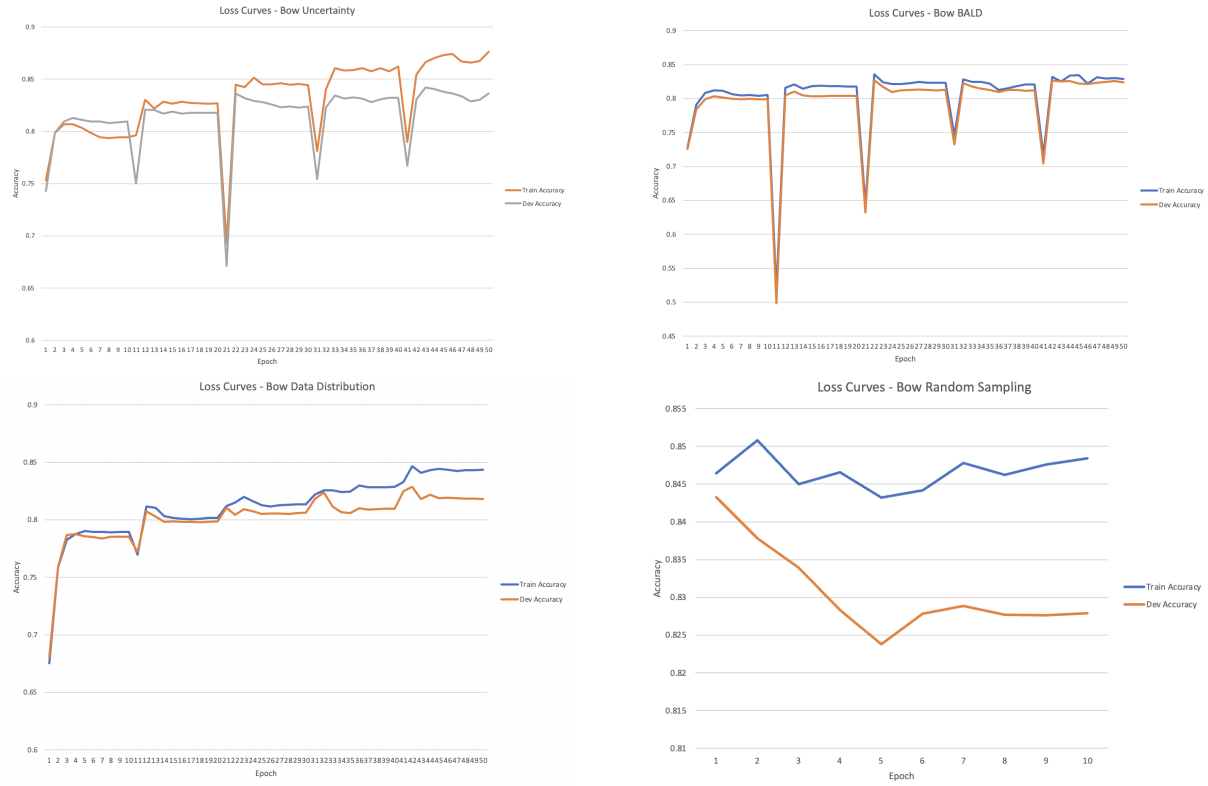
5.4 Results

After executing the code, we got the following accuracy results on the test set when comparing random sampling to BALD, uncertainty, and data distance.

Table 2: Accuracy on Test Set

Random Sampling	BALD	Data Distance	Uncertainty
82.79%	82.39%	81.80%	83.64%

The resulting loss curves of uncertainty, BALD, random sampling, and data distance learning methods are below.



Note that BALD and data distance strategies had less dev set accuracy than random sampling (82.39% and and 81.80% compared to 82.79%). Only the uncertainty strategy had greater accuracy than random sampling (at 83.64%). We analyze this discrepancy in the Analysis section.

The baseline model has an accuracy of 82.79% on the dev set, and only one active learning technique (Uncertainty) has a higher dev set accuracy of 83.64%. The other two active learning techniques (BALD and Data Distribution) have similar dev set accuracies as that of the baseline model but are slightly lower.

This indicates that although active learning techniques are effective at the sentiment classification task, they are not guaranteed to be better than the baseline random sampling method.

6 Analysis

To further analyze our results, we would like to determine if there are patterns of input movie reviews that are classified better by the baseline model and the three active learning technique models. We manually selected five movie reviews, which can be found in the Appendix.

Here are the sentiment prediction outputs of the baseline model and the three active learning technique models for the five manually selected movie reviews:

Table 3: Sentiment Prediction Outputs on Five Selected Movie Reviews

	Random Sampling	**BALD**	**Uncertainty**	**Data Distance**
Movie Review 1	Negative	Positive	Negative	Positive
Movie Review 2	Negative	Negative	Negative	Negative
Movie Review 3	Negative	Negative	Negative	Negative
Movie Review 4	Negative	Positive	Positive	Positive
Movie Review 5	Positive	Positive	Positive	Positive

Before we more thoroughly compare the models' sentiment predictions with each other, here is our subjective, human judgement on the sentiment of the five movie reviews:

Movie Review 1: Positive. This review is positive overall. However, it does have a few complexities by including key words like "I really liked" and also "... was boring and it has a terrible ending".

Movie Review 2: Negative. This review is clearly negative overall. It contains key words like "I didn't get", "I couldn't put my finger on [it]", and "... made no sense to me otherwise".

Movie Review 3: Negative. This review is clearly negative overall. It contains key words like "painfully bland" and "don't pick this up if you see it in a bargain bucket".

Movie Review 4: Negative. This review is negative overall. However, it does have a few complexities by including key words like "this was immensely boring" and "but it is cool to see..."

Movie Review 5: Positive. This review is positive overall. However, it does have a few complexities by including key words like "I can't help but notice the negative reviews" and "This movie is certainly worth your looking at".

For movie reviews 2, 3, and 5, which were all either clearly positive or negative, all models predicted the correct sentiment. For movie review 1, which is positive overall, only the BALD and data distance active learning techniques predicted it correctly. For movie review 4, which is negative overall, only the baseline model predicted it clearly. When the sentiment of the review is clearly positive or negative, then none of the models had difficulty in predicting sentiment. However, when the sentiment is more mixed or harder to discern from the review, there isn't clear evidence that active learning techniques perform better than the baseline model.

The BALD technique may be performing worse than the simpler uncertainity-based technique because BALD is more suited for deeper models. The dropout may be adverse affecting the small model's capacity to learn. Furthermore, all techniques may differ in their effectiveness depending on the portion of the dataset labelled in the learning process. In this set of experiments, we labelled 5000 of the 37500 training items, which represents 13.3% of the dataset.

Furthermore, it doesn't seem like an active learning technique or a specific model is better off at predicting sentiment according to the length of the movie review.

7 Conclusion

In completing this research, we aimed to experiment with active learning techniques to determine whether active learning could achieve better performances than a binary classifier used to classify movie reviews as having either positive or negative sentiment. To achieve this goal, we implemented three active learning techniques: cramming (uncertainty), curiosity-driven (BALD), and goal-driven (data distance). Essentially, these active learning techniques provided different ways to choose which sample from the training set to label next.

Using a dataset of IMDB movie reviews labeled with either positive or negative sentiment, we found that the baseline random sampling approach and the three active learning technique models all achieved similar accuracy results on the dev set. However, only one active learning technique (cramming/uncertainty) achieved a higher accuracy than the baseline model. This indicates that active learning can be effective, but it is not guaranteed to be better than a baseline sampling technique. On further analysis, we found that the baseline model and all active learning techniques were able to reliably predict the sentiment when the movie review was very clearly positive or negative. However, when the review was more nuanced and included both positive and negative connotation key words, then it becomes inconclusive which model or active learning technique performs better. In general, we were not able to determine a pattern amongst the inputs that corresponded to a best model/active learning technique to use. In the future, more in-depth analysis can be done beyond manually

sampling a few movie reviews to come up with a stronger conclusion for the best model/active learning technique to use for a certain input to optimize prediction accuracy.

A Appendix

Movie reviews selected for manual review mentioned in the Analysis section:

Movie Review 1:
"I really liked this Summerslam due to the look of the arena, the curtains and just the look overall was interesting to me for some reason. Anyways, this could have been one of the best Summerslam's ever if the WWF didn't have Lex Luger in the main event against Yokozuna, now for it's time it was ok to have a huge fat man vs a strong man but I'm glad times have changed. It was a terrible main event just like every match Luger is in is terrible. Other matches on the card were Razor Ramon vs Ted Dibiase, Steiner Brothers vs Heavenly Bodies, Shawn Michaels vs Curt Hening, this was the event where Shawn named his big monster of a body guard Diesel, IRS vs 1-2-3 Kid, Bret Hart first takes on Doink then takes on Jerry Lawler and stuff with the Harts and Lawler was always very interesting, then Ludvig Borga destroyed Marty Jannetty, Undertaker took on Giant Gonzalez in another terrible match, The Smoking Gunns and Tatanka took on Bam Bam Bigelow and the Headshrinkers, and Yokozuna defended the world title against Lex Luger this match was boring and it has a terrible ending. However it deserves 8/10"

Movie Review 2:
'Okay, I didńt get the Purgatory thing the first time I watched this episode. It seemed like something significant was going on that I couldńt put my finger on. This time those Costa Mesa fires on TV really caught my attention- and it helped that I was just writing an essay on Inferno! But let me see what HASŃT been discussed yet...

A TWOP review mentioned that Tony had 7 flights of stairs to go down because of the broken elevator. Yeah, 7 is a significant number for lots of reasons, especially religious, but hereś one more for ya. On a hunch I consulted wikipedia, and guess what Dante divided into 7 levels? Purgatorio. Excluding ante-Purgatory and Paradise. (The stuff at the bottom of the stairs and... what Tony cańt get to.)

On to the allegedly "random" monk-slap scene. As soon as the monks appeared, it fit perfectly in place with Tony trying to get out of Purgatory. You can tell he got worried when that Christian commercial (death, disease, and sin) came on, and heś getting more and more desperate because Christian heaven is looking kinda iffy for him. By the time he meets the monks heś thinking "hey maybe these guys can help me?" which sounds like contemplating other religions (e.g. Buddhism) and wondering if some other path could take him to "salvation". Not that Tony is necessarily literally thinking about becoming a Buddhist, but it appears Finnerty tried that (and messed up). That slap in the face basically tells Tony thereś no quick fix- as in, no, you cańt suddenly embrace Buddhism and get out of here.

Tony was initially not too concerned about getting to heaven. But at the "conference entrance", he realizes thatś not going to be so easy for him. At first I saw the name vs. driverś license problem as Tony having led sort of a double life, what with the killing people and sleeping around that he kept secret from most people. He feels free to have an affair with quasi-Melfi because "heś Kevin Finnerty". He figures out that he CAN fool some people with KFś cards, like hotel receptionists, but it wońt get him out of Purgatory. Those helicopters- the helicopters of Heaven?- are keeping track of him and everything he does.

After reading all the theories on "inFinnerty", though, it seems like KFś identity is a reminder of the infinite different paths Tony couldv́e taken in his life. Possibly along with the car joke involving Infinitiś that made no sense to me otherwise. Aaaand at that point my brain fizzles out.'

Movie Review 3:
'The first 30 minutes of Tinseltown had my finger teetering on the remote, poised to flick around to watch something else. The premise of two writers, down on their luck, living in a self-storage-space "bin" was mildly amusing, but, painfully bland.

The introduction of the character, played by Joe Pantoliano - the big deal movie guy, that lives in the park and sleeps in a lavatory, offered hope and I decided to give it a few more minutes. And then a few more until Kristy Swansons introduction as a budding film director borderline nymphomaniac, added a bit of spice. Her solid acting performance raised her presence above and beyond just a very welcome eye-candy inclusion.

Ultimately, the obvious low-budget impacts on the film with poorly shot

scenes, stuttured pace and slapstick handling of certain moments. Some of my favourite movies of all time have been low budget, Whithnail I being one that also deals with 2 guys with a dream, but down on their luck.

However, for my money, the actors save Tinseltown from the "Terrible movie" archives and just about nudges it into the "could have been a cult movie" archives. I laughed out loud at some of the scenes involving Joe Pantolianoś character. In particular, the penultimate scenes in the terribly clichéd, but still funny, rich-but-screwed-up characters house, where the story unravels towards itś final moments.

I can see how Tinseltown was a great stage play and while the film-makers did their best to translate this to celluloid, it simply didnt́ work and while I laughed out loud at some of scenes and one liners, I think the first 30 minutes dulled my senses and expectations to such a degree I would have laughed at anything.

Unless youŕe stuck for a novelty coffee coaster, donť pick this up if you see it in a bargain bucket.'

Movie Review 4:
'jeez, this was immensely boring. the leading man (Christian Schoyen) has got to be the worst actor i have ever seen. and another thing, if the character in the movie moved to America when he was ten or something and had been living here for over 20 years, he would speak a lot better English than what he pulls of here. or to say it in my own Language "Skikkelig gebrokkent". But it is cool to see Norwegian dudes in a movie made in Hollywood. it was just a damn shame they were talentless hacks. The storyline itself is below mediocre. I have a suspicion that Christian Schoyen did this movie just to live the dream, as he clearly does in the film by humping one beautiful babe after another.'

Movie Review 5:
"I can't help but notice the negative reviews this movie has gotten. To be honest, I saw the preview for this movie, and the premise looked intrigued me. Yes, I rented it after reading others' comments. They are correct in that some of the acting leaves a lot to be desired. They are also correct that one of the best performances of this movie was that of Dr. Graves.

Also interesting is Scott Clark, who plays Grant, the kid in the wheelchair. I identify with the character he played, perhaps because I am in a wheelchair.

This movie is certainly worth your looking at."

References

[1] Javad Bengar, Joost van de Weijer, Laura Fuentes, and Bogdan Raducanu. Class-balanced active learning for image classification. pages 1–9, October 2021.

[2] Patrick Hemmer, Niklas Kuhl, and Jakob Schoffer. Deal: Deep evidential active learning for image classification. pages 1–8, Portland, Oregon, USA, July 2020. International Conference on Machine Learning and Applications (ICMLA).

[3] Ksenia Konyushkovaand, Raphael Sznitman, and Pascal Fua. Learning active learning from data. pages 1–8, March 2017.

[4] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.