Determining Movie Similarities by Quantitative Text Analysis of Subtitles

Jan Luhmann
University of Leipzig
Computational Humanities
"Drama Mining and Film Analysis" course, summer 2019
jan.luhmann (at) gmx.net

Abstract

For streaming websites, media shopping platforms, movie databases and online communities, movie recommendation systems have become an important technology, where mostly hybrid methods of collaborative and content-based filtering on the basis of user ratings and user-generated content (tags, reviews) have proven to be effective. However, these methods can lead to popularity-biased results showing an under-representation of those movies for which only little user-generated data exists. In this paper we will discuss the possibility of generating movie recommendations not based on user-generated data or metadata but solely on the content of the movies themselves, confining ourselves to movies' speech. We extract speech-related low-level features from movies' subtitles using methods from the fields of Information Retrieval, Natural Language Processing and Stylometry, and examine a possible correlation of these features' similarity with the overall similarity of the respective movies. In addition we will present a novel web application for interactive evaluation of our experiment and for comparative analysis of extracted feature data.

Keywords: Movie Similarity, Subtitles Processing, Information Retrieval, Stylometry, Natural Language Processing

1. Introduction

With a rapidly increasing number of movies produced per year, a growing number of film industries worldwide[1] and new possibilities of distribution via streaming websites such as Netflix and Amazon Prime, and on-demand services like Vimeo and Youtube Movies, recommendation systems for movies have become essential for the users' experience. Users who are eager to discover and watch movies unknown to them would be completely lost at the attempt to single-handedly pick the ones they are interested in from the mass of released movies.

Movie recommendation systems which are being used currently are largely based on collaborative filtering, which means that recommendations largely depend on two factors: Which movies does a particular user like and which not? Which users like a particular movie and which do not? (Bennett and Lanning, 2007; Su and Khoshgoftaar, 2009)

A major challenge for collaborative filtering recommendation systems is the so-called cold start problem. A cold start, i.e. the case that a movie does not yet have enough user ratings or that a user did not yet rate enough movies to calculate any recommendations using collaborative filtering, is mostly handled using content-based approaches: Using metadata provided by the movie distributor such as genre tags, list of cast and plot synopsis, as well as tags (regarding plot, style and mood of a movie) and reviews provided by users, a basis of movie similarities can be determined which is then used for initial recommendations (Sarwar et al., 2001).

However, recommendation systems using this approach still suffer from a popularity bias: Any movie which has not been sufficiently tagged by users or whose metadata is only fragmentary is a lot less likely to survive a cold start. To increase diversity and novelty in movie recommendations, it would be greatly beneficial to be able to estimate movie similarities independently of any human-supplied attributional data but based on the content of movies themselves.

In addition of the actual video and audio data of a movie, a third resource which can represent one aspect of a movie, i.e. its dialogue, is its subtitles. Of course, subtitles can only contain a fraction of the information which a movie's dialogue provides, completely missing information about speakers, intonation, facial expressions etc.

But since it still may contain information about dialogue topics and manner of speaking, and since English subtitles are widely available today, even for very obscure movies, through online platforms such as *OpenSubtitles*[2], and can be processed inexpensively and efficiently in comparison to video or audio data, we will here discuss and explore the possibility of detecting movie similarities using feature extraction from subtitles. The applied methods of feature extraction are related to Natural Language Processing (NLP), Information Retrieval (IR) and Stylometry.

1.1. Related Work

Before we present the experimental setup, we discuss a number of works that use subtitles and movie scripts as data sources in the context of determining movie similarities.

In their 2008 paper, Alex Blackstock and Matt Spitz propose a method for classifying 399 movies by NLP-related features extracted from movie scripts. Their method is partly stylometric, examining the ratios and distributions of occurrences of grammatical word forms using Part-of-Speech tagging (POS), partly statistical using features derived from speaker annotations which are present in movie scripts, and partly using Named Entity Recognition (NER) for analysis of identical named entities. Movies are classified by genre using MEMM and Naive Bayes. While stylometric features achieve the best results, the overall accuracy is relatively low. The authors conclude that a larger and more diverse dataset would have improved their results. However, freely and digitally available movie scripts are much rarer than subtitle files and more difficult to process.

[1] UNESCO Institute of Statistics. (2016) Record number of films produced. `http://uis.unesco.org/en/news/record-number-films-produced` (date accessed: 2019-08-29)

[2] `http://www.opensubtitles.org`

Barbara Cimpa and Jochen Nessel (2011) propose a movie recommendation engine which uses a Inductive Inference-based method of calculating similarities among subtitle texts of 290 pre-selected movies. The results of the evaluation experiments look promising, although it is difficult to say how their approach would perform on a more diverse dataset.
In 2017, Konstantinos Bougiatiotis and Theodoros Giannakopoulos examine the correlation of movie similarities and features extracted from subtitles and audio, using a dataset of subtitle files of 160 movies. Bag-of-Words (BOW) representations of subtitle text are used for calculating topic models. Segments of audio data are classified by event (music, speech, noises etc.) and in case of music classified by music genre. In evaluation, extracted audio features only yield very low accuracy scores. The two most accurate results are generated by topic modelling using Latent Dirichlet Allocation, and by simple tf-idf weighting of BOW representations.

2. Experimental Setup

In this section we introduce the dataset employed for our experiment and the methods we applied for feature extraction, for scaling and normalizing our feature models, and for similarity measurement.

2.1. Dataset

Our dataset consists of English subtitles for 5914 movies. These movies are all among the 10000 most rated movies on *IMDb*[3]. Despite our motivation to tackle a popularity bias, we choose well-known movies to better being able to assess the quality of our experiment. We choose such a large and diverse dataset because it may improve the quality of some models and more accurately represent a later real-world application.
Subtitles are kindly made available by *OpenSubtitles*. For each movie they provide us with several versions of subtitles. Often subtitles contain OCR errors (optical character recognition), encoding errors, and for our purposes unwanted data like speaker annotations, music lyrics, authorship tags by the subtitle creator and HTML markup. To minimize the occurrence of such data, subtitles for all movies undergo a rudimentary cleaning, validation and selection process (see Section 2.1.2.), which leaves us with one selected subtitle file for each of 5914 movies, formatted as a SubRip file (*.srt).
Additionally, metadata (*IMDb*-ID, title, release year, genres, runtime, number of ratings, rating) for these movies are obtained from *IMDb*.

2.1.1. SubRip File Format

SubRip is a popular file format for subtitles and originates from the software SubRip by developers Brain and Zuggy[4], which is used to extract hard-coded subtitles from video data using OCR. The format is stored as plain text, an encoding standard does not exist.

```
695
00:57:27,891 --> 00:57:29,256
He said " phone''?

696
00:57:29,326 --> 00:57:31,590
Can't you
understand English?

697
00:57:31,661 --> 00:57:33,185
He said phone.

698
00:57:36,500 --> 00:57:38,593
Home...
```

Figure 1: Excerpt from subtitle of "E.T. the Extra-Terrestrial" (1982)

The format follows this structure:

1. Sequential count of subtitles, starting with 1.
2. Start timecode, `" --> "`, end timecode.
   Timecodes are formatted as HH:MM:SS,MIL (hours, minutes, seconds, milliseconds).
3. Dialogue lines (min. 1).
4. Empty line.

2.1.2. Subtitle Cleaning and Validation Process

In a first step, subtitle files are being parsed into a table format (list of lists), separating line count, timecodes and dialogue lines (see Section 2.1.1.).
In a second step, the dialogue lines are cleared of:

- any text within brackets (unlikely to be spoken text)
- any HTML markup
- any "unspeakable" characters (rare characters and symbols which don't have a verbal representation in the English language)
- speaker annotations at the beginning of lines (only occurs rarely)
- dashes or ellipses (multiple dots) at the beginning of lines
- dashes or ellipses at the end of lines (replaced with single dot)
- any authorship tags by the subtitle creator within the first and last 5 dialogue lines (keywords such as "sub", "sync", "rip", etc.)

In a third step, the cleaned subtitle is validated. It has to satisfy the following criteria which we set rather intuitively based on observations on our dataset:

1. The dialogue has at least 1000 characters.
2. At least 25% of the 178 words in the *NLTK*[5] stopwords list occur among the first 300 words of the dialogue.
3. The sequence count number of the last dialogue line and the actual total number of subtitle lines differ by no more than 15.
4. There is dialogue within the first 20 minutes.
5. The movie runtime (obtained from *IMDb*) and the end timecode of the last dialogue line differ by no more than 30 minutes.
6. The ratio of the total number of subtitle lines to the movie runtime (in seconds) is larger than 1%.

[3] `http://www.imdb.com`

[4] `https://en.wikipedia.org/wiki/SubRip` (date accessed: 2019-08-29)

[5] Natural Language Toolkit: `https://www.nltk.org/`

If a subtitle file for a particular movie does not match these criteria, another subtitle file for this movie is validated. If no subtitle file for a movie has proven valid, the movie is not included in our dataset.
In Section 3, results of our experiment will show that our intuitively set validation criteria are not sufficient in preventing subtitle files which contain OCR errors from occurring in our dataset. These errors are not addressed more explicitly during validation process simply because it is only during evaluation that we will realize how often OCR happens to be the source of subtitle files.
We will discuss possible improvements to the validation process in Section 4.

2.1.3. Preprocessing

For some of our feature extraction methods which are described in Section 2.2. it is necessary to convert a movie's subtitles into a continuous text. This is done by simply joining all dialogue lines to a single string, separated by whitespace. The text is then further processed using *spaCy*'s language processing pipeline [6]. In our case, this pipeline consists of the following stages: tokenization, POS tagging, dependency parsing, NER, lemmatization. For each movie, we store:

- the sequence of tokens
  - lowercase
  - without punctuation
  - named entities of types `PERSON`, `ORG` (Organization) and `WORK_OF_ART` filtered out
- the sequence of lemmatized tokens
  - lowercase
  - without punctuation
  - named entities of types `PERSON`, `ORG` (Organization) and `WORK_OF_ART` filtered out
  - all personal pronouns are replaced by the token "PRON"
- the sequence of POS tags corresponding to tokens
  - including sentence start/end markers ("#")

An explanation for filtering out named entities is given in Section 2.2.1.

2.2. Methods for Feature Extraction

In this section the applied methods for feature extraction are presented. Their configurations and the adjustments of their parameters were determined experimentally, i.e. by testing them on randomly selected samples of our dataset and examining the results.
There is one method for topical analysis of documents, in this case with the motivation to address the question: *What are the characters of a movie talking about?* (see Section 2.2.1.). Four methods are aimed at stylistic analysis, in an attempt to address the question: *How are the characters of a movie talking?*, considering aspects of lexicality, syntax and speech rhythm (see Sections 2.2.2., 2.2.3., 2.2.4, 2.2.6). A sixth method is aimed at an analysis of emotions: *Which emotions are the characters expressing in their speech?* (see Section 2.2.5.)

[6] `http://www.spacy.io/`

2.2.1. Bag-of-Words Model (BOW)

The Bag-of-Words model is a simple approach for representing text documents in Information Retrieval and Natural Language Processing. In our model, the subtitle text for a movie (a document) is represented by its set of unigrams of lemmatized tokens, weighted by sublinear-tf-idf scaling (Manning et al., 2008, p. 126-127) which is a logarithmic variation of tf-idf scaling (term frequency - inverse document frequency).
By this weighting those terms (here: unigrams) which occur frequently in a certain document, but occur altogether in very few documents of our dataset, appear as the most significant terms of this document. Ideally, these can be interpreted as the most significant terms to the documents' topics. On the other hand, common terms which occur in this document but also occur in mostly all documents are weighted much lower due to the factor of inverse document frequency.
By logarithmizing the term frequency factor, the actual number of occurrences of a term in the document has a less drastic effect on the weighting.

$$\text{sublinear-tf-idf}(t,d) = \text{sublinear-tf}(t,d) \cdot \text{idf}(t) \quad (1)$$

$$\text{sublinear-tf}(t,d) = \begin{cases} 1 + \log \text{tf}(t,d) & \text{if } \text{tf}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where:

$\text{tf}(t,d)$: frequency of term in document
d : document
t : term

$$\text{idf}(t) = \log \frac{1+N}{1+n_t} + 1 \quad (3)$$

where:

N : total number of documents
n_t : number of documents where *t* occurs
t : term, here: unigram

In order to filter out the overall most common terms which may not have any semantic significance to a document (stop words) and also to reduce dimensions of our document representation, all terms which occur in more than 95% of all documents are ignored. This limit is adjusted to the occurrence of "traditional" stop words. Likewise, all terms which occur in less than 2.5% of all documents are ignored. This limit is adjusted to the occurrence of terms containing spelling mistakes.
And as mentioned in Section 2.1.3., all tokens which have been recognized as named entities of types `PERSON`, `ORG` (Organization) and `WORK_OF_ART` are filtered out during preprocessing, since otherwise movies featuring identically named characters would contain these characters' names as very significant terms in their respective subtitle documents. This would ultimately result in movies appearing similar only because of these names. We consider the

types ORG and WORK_OF_ART because frequently characters' names are falsely categorized as these.
Finally remaining are 5240 terms, so that documents are represented by this model as vectors of dimension 5240.

2.2.2. Distribution of Stop Words (SWD)

One method that has long been successfully used in stylometry, particularly in studies of author attribution, is to measure the rates of occurrences of stop words in a given document and to understand the distribution of these as a fingerprint of the author's writing style. This was mainly developed by John Burrows during the late 1980s.
We will apply this method on our dataset of unlemmatized tokens. In a first step, we split all tokens of contracted words found in our dataset such as doesn't, won't or would've into their morphemes, does, n't, wo, n't, would, 've by using *NLTK*'s tokenization. Now it is possible to measure these morphemes as separate tokens. This task is not done natively by *spaCy*'s tokenization.
We generate a set of 94 stop words by taking *NLTK*'s list of English stop words, extending it manually with the most common stop words found in our dataset and filtering out stop words not occurring in our dataset.
Document representations are then computed in the same way as our Bag-of-Words-Model but considering only the terms of the stop word set, weighted simply by their term frequency.

2.2.3. POS Tag Trigrams (PTT)

Commonly used for genre classification of text documents is the method of extracting features related to syntactic structures. Genres in this case are usually fiction, academic text, news text, conversation, etc., where the syntactic structure is a discriminating factor between genres, regardless of the topic of texts. This method was used by Argamon et al. (2003) to classify texts by their author's gender, and by Santini (2004) for genre classification. As features they used trigrams of POS tags which "are large enough to encode useful syntactic information, and small enough to be computationally manageable" (Santini, 2004), so we choose to do the same in this work.
The POS tags emitted by *spaCy* are Pennbank tags.[7] During preprocessing, we have added sentence start/end markers and we will only allow these occurring at the beginning or end of a trigram, so trigrams represent sentence-bound structures only. Examples of possible trigrams are: '#_DT_NN', '#_UH_#', 'DT_NN_VBD', 'IN_JJ_NN', 'MD_VB_#'.
Similarly to the method presented in Section 2.2.2., we are more interested in the distribution of frequencies of common features than finding significant rare features of our documents. Also, valid syntactic structures of English are limited and we cannot be sure of the level of correctness of POS tagging. For these reasons we ignore all trigrams that occur in less than 90% of our documents, resulting in 429 trigrams. Weighting is done by tf-idf scaling (see Section 2.2.1.) to balance out the general frequencies of occurrence of trigrams in our documents.

2.2.4. Stylometric & Statistical Measurements (SSM)

Next we calculate a model of document representations based on various ratios and measurements which are popular in stylometric studies and which are once again often successfully used for genre classification and authorship attribution. These include:

- Entropy:

$$\mathrm{H}(d) = -\sum_{t \in d} \mathrm{P}(t) \cdot \log \mathrm{P}(t) \tag{4}$$

  where:

  d : document
  t : term
  $P(t)$: probability of t occurring in d, $\frac{\mathrm{tf}(t,d)}{|d|}$

  Entropy was introduced by Claude Shannon in 1948 who defined it as a measure of "choice and uncertainty". In linguistic context, it can be interpreted as a measure of lexical diversity.

- Standardized type-token ratio:
  Another more basic measure of lexical diversity, calculated by dividing the number of a document's types by the total number of tokens. As this measure is very sensitive to document length and our documents are very different in length, we use a simple standardized variant here called Mean Segmental Type-Token Ratio: Each document is divided into segments of 1000 tokens and for each segment a type-token ratio is calculated. If a last segment of less than 1000 words remains, this is excluded. The mean value of all these segments' ratios is the standardized type-token ratio of the document (Johnson, 1944; Torruella and Capsada, 2013).

- Sentence lengths:
  For each document, the mean value of sentence lengths (words per sentence) is calculated and furthermore the ratios of the number of short sentences (less than 5 words), medium-sized sentences (5 to 7 words) and long sentences (more than 7 words) to the total number of sentences. These measurements could complement our PTT model by approximately capturing the syntactic complexity of a document.

- Word lengths:
  Likewise we calculate the mean value of word lengths (characters per word) as well as ratios of the number of short words (less than 4 characters), medium-sized words (4 to 5 characters) and long words (more than 5 characters) to the total number of words. Hereby the lexical complexity of a document may be roughly estimated.

Further added features to this model, which are not directly related to stylometry, are mean values of sentiment, speech tempo, speech pause durations and a speech pause ratio. These are described and explained as part of the following two sections.

[7] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html (date accessed: 2019-08-29)

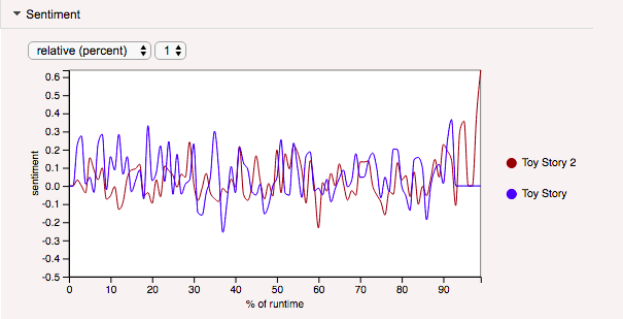

Figure 2: Detail view of Sentiment Analysis curves in evaluation tool *Sub Rosa* (see Section 2.4.)

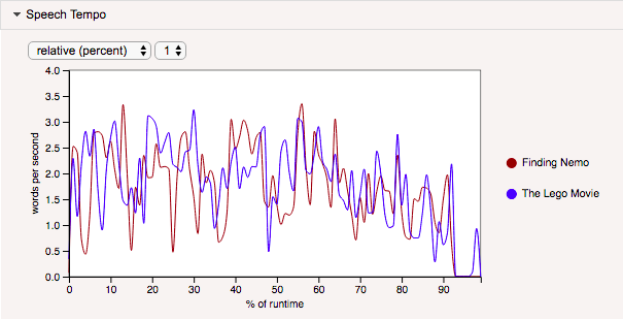

Figure 3: Detail view of Speech Tempo Analysis curves in evaluation tool *Sub Rosa* (see Section 2.4.)

2.2.5. Sentiment Analysis (SEA)

By sentiment analysis of a document of our dataset we may estimate the emotional arc of the respective movie, assuming that this arc is in some way reflected in the dialogue. We use the well evaluated open-source tool *VADER Sentiment*[8] (Hutto and Gilbert, 2014) for calculating a *compound sentiment score* for a given text, which is positive (> 0 and ≤ 1) in case of a text that reflects positive sentiment and negative (< 0 and $\geq$ -1) in case of a text that reflects negative sentiment.

Since emotions usually evolve very intensely over the course of a movie, it is not useful to calculate the sentiment of an entire document of movie dialogue at once. Instead we divide each movie's dialogue into segments of 1% of its runtime using the subtitles' timecodes. Then for each of these segments, the sentiment of dialogue is calculated, resulting in a sentiment curve for each movie (see Fig. 2). The first two segments and the last three segments are excluded because these often coincide with movies' opening and closing credits. For better comparability, smoothing using a simple moving average (window size of 5) is applied to the curve. We store it as a vector of 95 dimensions, each representing a segment's sentiment score. Also, the mean value is calculated and appended to our statistical model of Section 2.2.4.

2.2.6. Speech Tempo Analysis (STA)

It is intuitively understandable that distinctive features of a movie may be the tempos at which characters speak as well as the frequency and duration of speech pauses. These may correlate with the rhythm of a movie's editing which is crucial to its style and atmosphere (Van Leeuwen, 1985). Using the same document segments (each 1% of runtime) generated for sentiment analysis, it is also easily possible to analyze the speech tempo of a movie. For each segment, we calculate the speech rate by dividing the number of spoken words by the actual duration of the segment (in seconds), creating a curve of speech rates over the course of a movie (see Fig. 3). Here as well, the first two segments and the last three are excluded and smoothing by simple moving average (window size of 5) is applied to the curve.

Furthermore, we calculate the mean value of speech rates. We also measure the mean value of the duration of speech pauses (> 10 milliseconds), with the exception of those that occur at the beginning and end of a movie, as these can be caused by opening or closing credits. This data is then also used to calculate the ratio of the summarized duration of speech pauses to the runtime of the movie.

The mean value of speech rates, mean value of the duration of speech pauses and the speech pause ratio are appended to our statistical model of Section 2.2.4.

2.3. Similarity Measurement

In this work, our focus is on an efficient, dynamic and transparent examination of the similarities of films regarding their extracted features. This will become more apparent in Section 2.4. with the introduction of our web application.

Therefore, the approach chosen for the similarity measurement is quite simple: Vectors are scaled and normalized. Then for each movie, all its feature vectors are weighted and concatenated. Between concatenated vectors (representing different movies) the Euclidean distance is calculated.

2.3.1. Scaling & Normalization

The vectors produced by the BOW, SWD and PTT models are normalized to ℓ^2 unit norm, which compensates for different document lengths and prepares for later concatenation.

In our SSM model each individual feature is scaled globally by min-max scaling. Vectors produced by SEA model are min-max scaled to avoid negative values.

For these two models and the STA model, it is essential to consider the absolute magnitude of vectors for similarity measurement, since it is not only the distribution of values that is decisive, but also their actual value. By simple ℓ^2-normalization, we would lose this information. On the other hand, for later concatenation and weighting of feature models, it would be useful for vectors to be of unit norm.

We work around this problem by calculating for each of the models a global mean vector, determining its normalization factor to scale it to ℓ^2 unit norm, and subsequently normalizing all vectors of the model by this factor. As a result, all vectors of the model are approximately of unit norm while keeping their relations of magnitude. Some slightly smaller than unit norm, some slightly larger.

This solution is certainly neither the most elegant nor the most mathematically accurate, but at the moment it will al-

[8] `https://github.com/cjhutto/vaderSentiment` (date accessed: 2019-08-29)

low us to study the results of our experiment pragmatically and without major distortions.
By normalizing all vectors to the approximate unit norm, we ensure that their different numbers of dimensions (coming from different models) do not affect their impact on later similarity measurement.

2.3.2. Weighting & Concatenation

It is yet unknown how the features of our different models should be weighted in order to achieve results of similarities that correlate most closely with a general similarity of movies. In this work we leave it to the user of the web application to experiment with the weighting of the features (see Section 2.4.).
The individual feature vectors of any movie are weighted according to the user's adjustments and then concatenated into one. Each concatenated vector representing a movie has 5967 dimensions, of which 5240 dimensions stem from the BOW model, 94 dimensions from SWD, 429 dimensions from PTT, 95 dimensions from SEA, 95 dimensions from STA, 14 dimensions from SSM model.

2.3.3. Distance Metric

As explained in Section 2.3.1, it is relevant here to consider vector magnitudes when calculating distances. Due to this condition Euclidean distance is chosen as the distance metric, even though it is usually not recommended for high-dimensional and sparse data as it is present here. Because of the *curse of high dimensionality* (Aggarwal, 2001) all our vectors are approximately equidistant from each other by Euclidean metric (distances of $\sim\sqrt{2}$). As results will show, however, we are nonetheless able to retrieve meaningful rankings of nearest neighbors. A further discussion on this issue will take place in Section 4.

2.4. Web Application for Interactive Evaluation

We present a novel web application called *Sub Rosa*, designed for interactive evaluation of the results of our experiment: `http://github.com/bbrause/subrosa`. It allows users to adjust the weighting of feature models based on which distance calculations are performed. Users can have the nearest neighbors of any movie calculated and can also compare individual movies' extracted features in detail. Similarity measurements are not pre-calculated, but are performed dynamically.
Detail views of the data of SEA and STA are enhanced by additional data for variably customizable time windows (segment durations, either relative to runtime or absolute in minutes). These, plus the detail view of SSM display values before scaling and normalization, giving users the ability to interpret them. This data in this form is not used in calculations, but was stored solely for presentation purposes during feature extraction.
The values of similarity between movies are visualized in a graph (see Fig. 4) in which each node represents a movie and the length of the edge between each two movies is proportional to the square of the Euclidean distance calculated between them.

3. Results

We can estimate whether the modelling leads to meaningful results regarding our research question by analyzing two-dimensional projections of all vectors of the models. These were made by first reducing to 50 dimensions using Truncated SVD[9] and then to two dimensions using t-SNE for visualization (van der Maaten and Hinton, 2008). It should be noted that t-SNE does not retain original distances between vectors, but can be influenced to exaggerate clustering. We configure the parameters of t-SNE in such a way that natural clusters are emphasized, but no artificial clusters are formed. The data points of our projections are colored according to the genre of the movie they represent.
Let's take a look at Fig. 5, showing a projection of concatenated vectors of all feature models. Here, vectors were not weighted before concatenation, so that each model has the same influence. What is most striking is that, clearly separated from the main cluster of points, there is a small cluster that floats above it. After some investigation within our dataset, we find out that this phenomenon is caused by OCR errors. Each of the points within this small cluster originates from a subtitle file containing typical incorrectly recognized words such as `lt`, `lf`, `ls`, `Iook`, `Iike` (lowercase `l` mistaken for uppercase `I` and vice versa, see Fig. 16-17). This especially affects the BOW, PTT and Stopwords Distribution models, as can be seen clearly in their individual 2D visualizations (Fig. 9-11).
The distribution of the colors of the points in Fig. 5 shows distinct large accumulations of yellow points (Comedy) and black points (Horror) as well as widely scattered smaller accumulations of red and green points (Drama and Adventure), indicating similarities of features among the respective movies. A detail view of the accumulation of Horror movies (Fig. 6) shows tiny clusters of movies which are loosely related by plot or setting. It also shows some movies which seem out of place (such as "Dinosaur" (2000) or "The Good Dinosaur" (2015) which both are children- or family-oriented movies).
If we exclusively examine a combination of the BOW, SEA and SSM models (Fig. 7), we notice much more distinct clusters and accumulations. Each cluster appears to unite movies with a specific setting or style, not restricted to a particular genre. In detail (Fig. 8), tiny clusters of movies with even more specific settings can be seen.
The 2D projections of each model individually (Fig. 9-14) show that modelling by BOW, PTT, SWD and SSM produce rather clustered representations while SEA and STA produce evenly distributed representations. For the last two, it is remarkable that while other genres are widely scattered, there is still a comparatively clear distinction between Comedy and Horror movies.
A more detailed evaluation of the quality of our models will be possible if a ground-truth dataset of human-estimated similarities of movies is available to us. For now, the reader is kindly invited to explore the results with the help of *Sub Rosa*, available on our *GitHub* repository. All 2D projections are also available there as interactive plots.

[9] `https://scikit-learn.org/stable/modules/decomposition.html` (date accessed: 2019-08-29)

4. Conclusions

First of all, we discuss some improvements to be made. Since our results are partly corrupted by OCR errors, the validation process (as described in Section 2.1.2.) must be enhanced to detect files that contain such errors. This can be achieved by scanning the files for occurrences of the most common errors. Techniques of correcting such errors may also be considered.

For some movies, especially children-oriented movies, the most significant terms in their representation by BOW model seem to be various interjections, which makes them appear similar to other movies containing similar interjections (see Fig. 15). This wouldn't exactly be a problem if the idiosyncratic ways of spelling interjections weren't more a feature of the writing style of a subtitle author than of the dialogue of a movie. A solution to this problem would be to filter out interjections during preprocessing by the POS tag 'UH' or replacing them with a placeholder. Alternatively, it would be conceivable to allow only nouns (and possibly verbs) as terms of the BOW model, as it is sometimes done in topic modelling (Martin and Johnson, 2015). This approach could also be a solution to the problem of OCR errors if given that the incorrect words are not recognized by *spaCy* as nouns.

Regarding the issue of the *curse of high dimensionality* (see Section 2.3.3.), it would be most advisable to perform distance calculations individually for each model instead of first concatenating their vectors. A compound distance score would then be calculated from the individual distance values. This process would, however, lead to a more complex backend of *Sub Rosa*.

In the case of the BOW model dimension reduction methods such as LDA (Blei, 2003) may be applied to vectors prior to distance calculation, effectively resulting in a topic model. Whether this will lead to accurate results would have to be evaluated. For SEA and STA models, it would be possible to calculate distances on their curve data using a more appropriate and sophisticated method than Euclidean distance (Efrat et al., 2007).

A possibility for proper evaluation of our experiment may be provided by the "MovieLens" dataset by the research lab *GroupLens* (University of Minnesota, USA)[10]. It offers user-generated tags applied to 27000 movies as well as relevance scores to these tags. Tags mostly refer to style, mood, plot or setting of a movie. Similarities of movies regarding their tags may be compared with their similarities regarding our models.

Beyond evaluation, future work may use the "MovieLens" and our dataset to explore a possibility of using machine learning algorithms to learn to automatically apply multiple tags as well as genres to any movie solely based on its subtitle data.

5. Acknowledgements

The author is very grateful to the *OpenSubtitles* team for providing part of their subtitle database which made this work possible in the first place.

This work was realized as part of the course "Drama Mining und Film-Analyse" (summer semester 2019) under the supervision of Manuel Burghardt and Jochen Tiepmar at the University of Leipzig.

[10] http://grouplens.org/datasets/movielens/ (date accessed: 2019-08-29)

6. Bibliographical References

Aggarwal, C. C. (2001). On k-anonymity and the curse of dimensionality.

Argamon, S., Shimoni, A. R., and Koppel, M. (2003). Automatically categorizing written texts by author gender.

Bennett, J. and Lanning, S. (2007). The Netflix prize.

Blackstock, A. and Spitz, M. (2008). Classifying movie scripts by genre with a MEMM using NLP-based features.

Blei, D. M. (2003). Latent dirichlet allocation.

Bougiatiotis, K. and Giannakopoulos, T. (2017). Multimodal content representation and similarity ranking of movies.

Burrows, J. (1987). Computation into criticism: A study of Jane Austen's novels and an experiment in method.

Efrat, A., Fan, Q., and Venkatasubramanian, S. (2007). Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. 27(3).

Hutto, C. J. and Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text.

Johnson, W. (1944). Studies in language behavior.

Manning, C., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Martin, F. and Johnson, M. (2015). More efficient topic modelling through a noun only approach.

Nessel, J. and Cimpa, B. (2011). The MovieOracle - content based movie recommendations.

Santini, M. (2004). A shallow approach to syntactic feature extraction for genre classification. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics (CLUK 2004)*.

Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international conference on World Wide Web - WWW '01*, pages 285–295. ACM Press.

Shannon, C. (1948). A mathematical theory of communication.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques.

Torruella, J. and Capsada, R. (2013). Lexical statistics and tipological structures: A measure of lexical richness.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE.

Van Leeuwen, T. (1985). Rhythmic structure of the film text. In Teun van Dijk, editor, *Discourse and Communication: New Approaches to the Analysis of Mass Media Discourse and Communication*, pages 216–232. Walter de Gruyter.

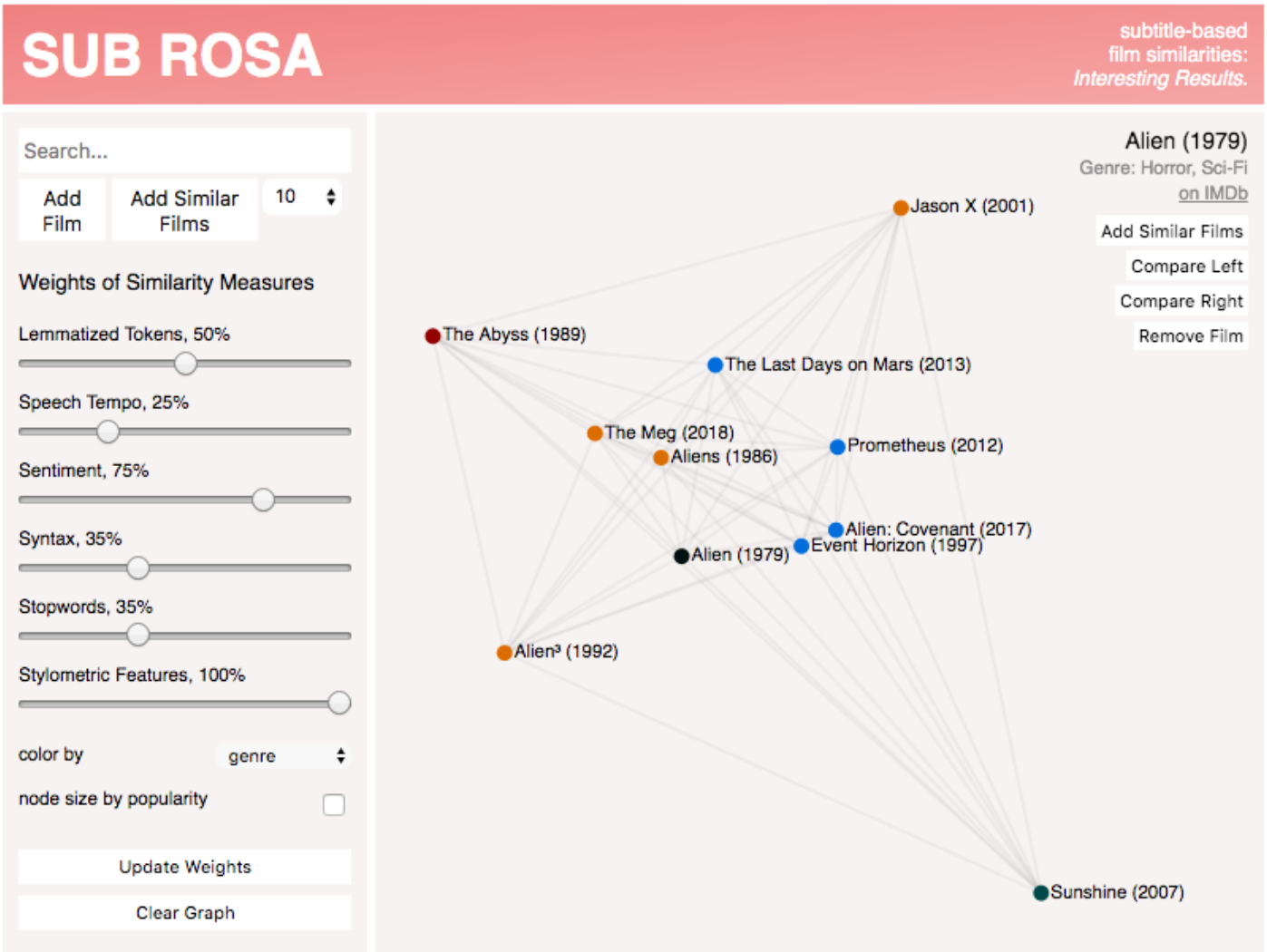

Figure 4: Main interface of *Sub Rosa* displaying ”Alien” (1979) and similar movies

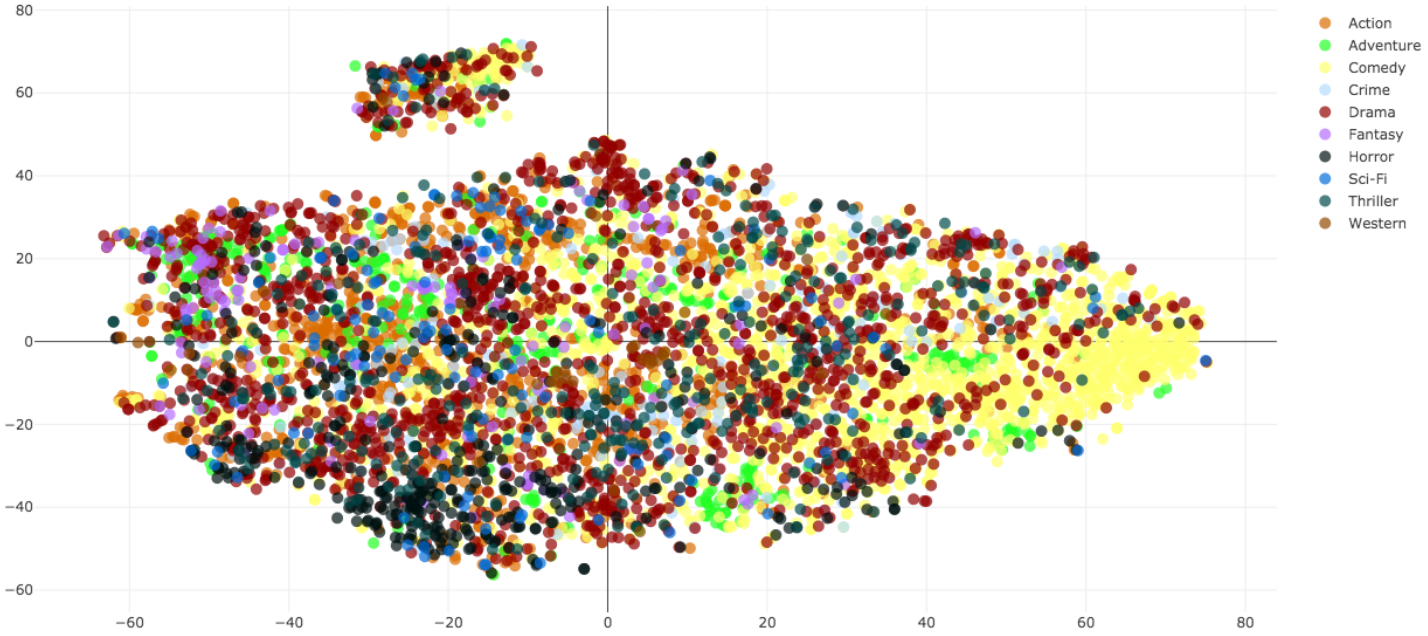

Figure 5: All feature models concatenated (unweighted), 2D projection using Truncated SVD and t-SNE

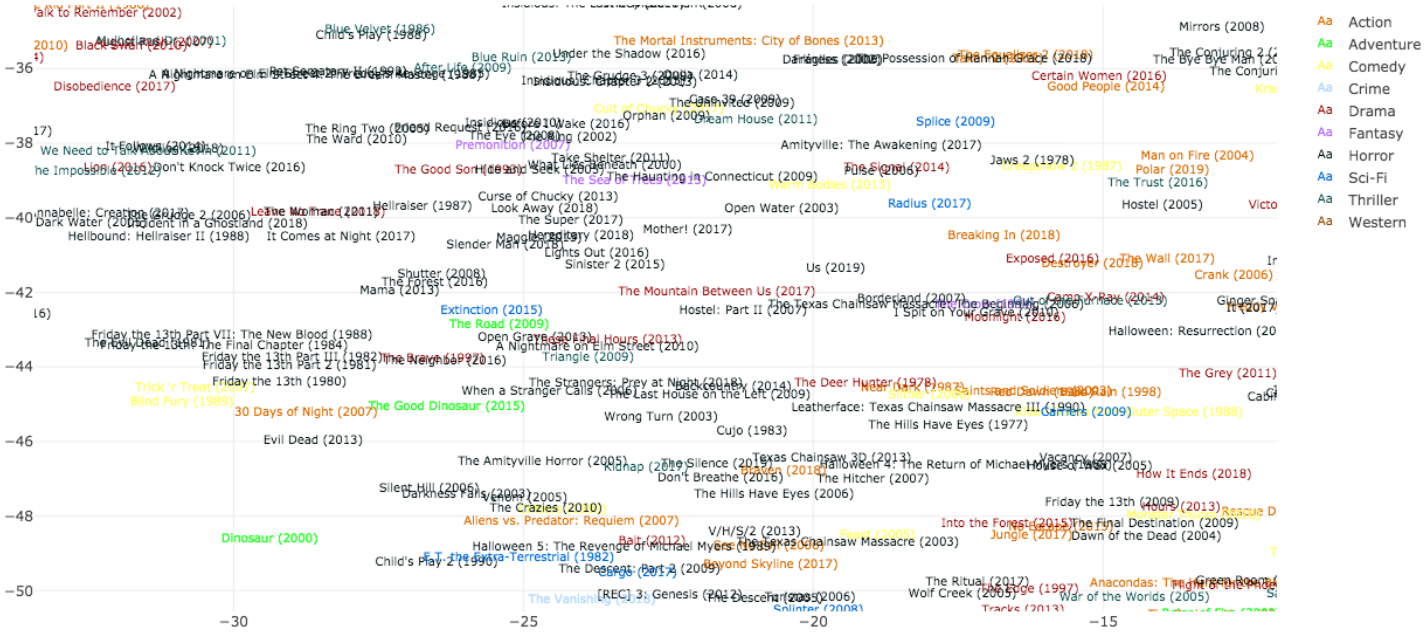

Figure 6: Mostly horror movies (detail of the projection from Fig. 5)

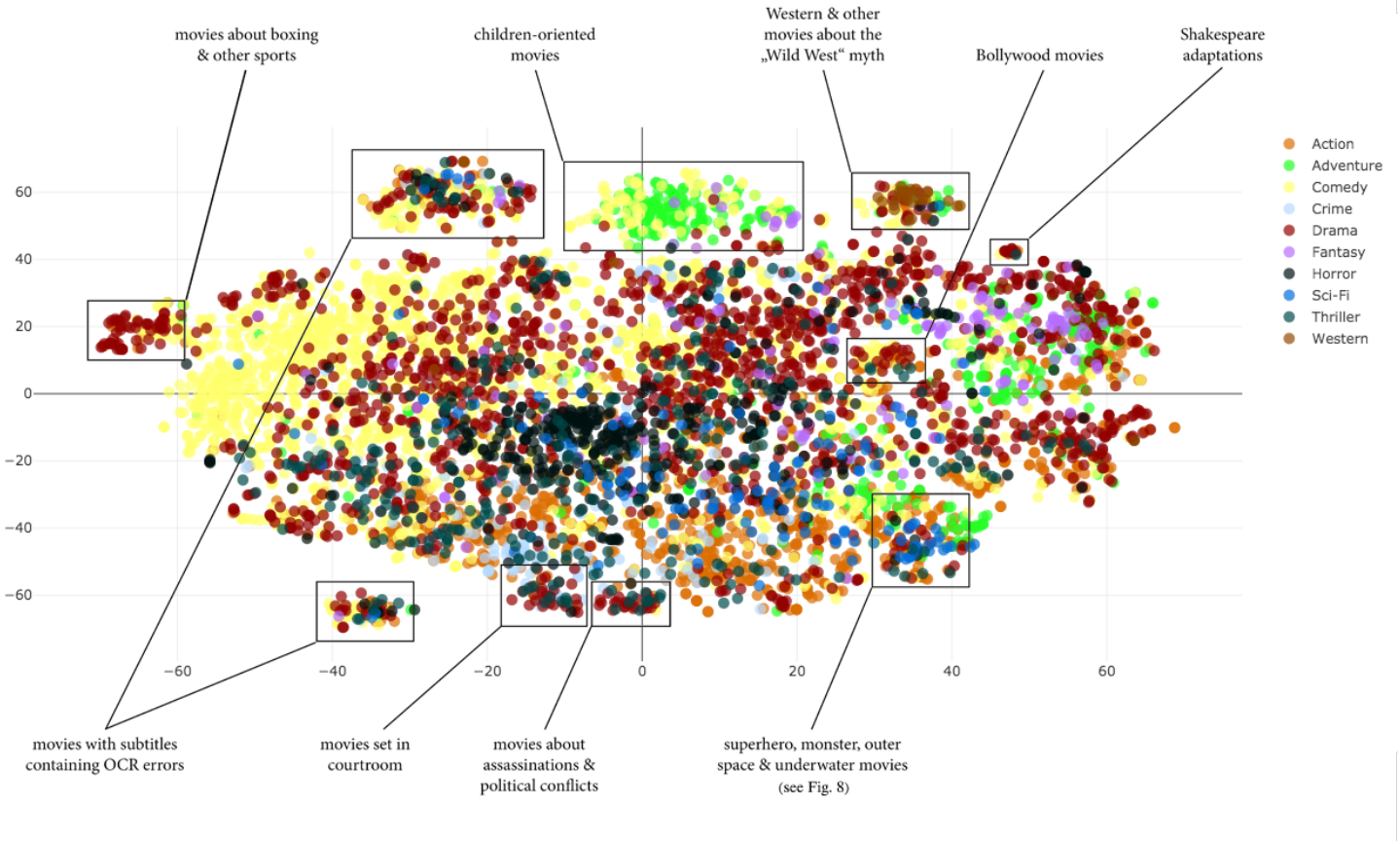

Figure 7: Bag-of-Words, Sentiment Analysis & Stylometric Measurements models combined (unweighted), 2D projection using Truncated SVD and t-SNE, with annotations of a few distinctive clusters

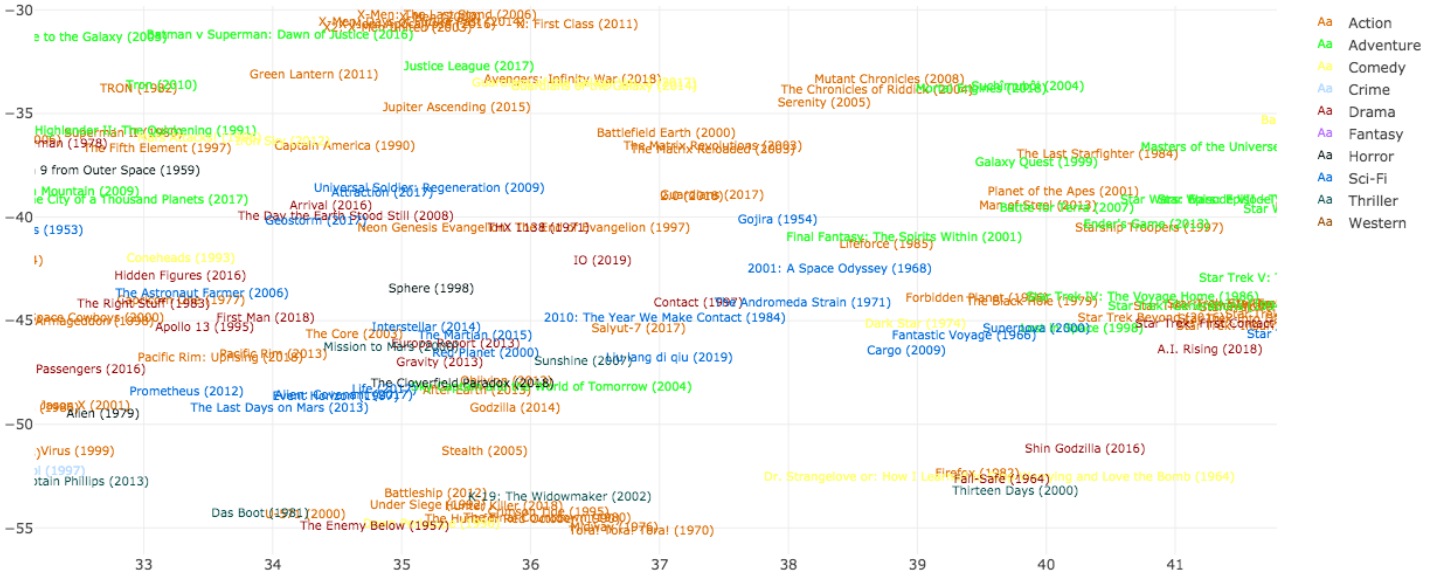

Figure 8: Superheroes, monsters, outer space & underwater (detail of the projection of Fig. 7)

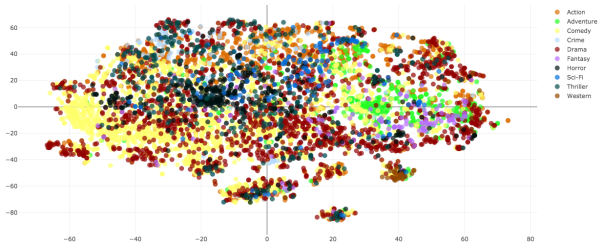

Figure 9: Bag-of-Words model, 2D projection using Truncated SVD and t-SNE

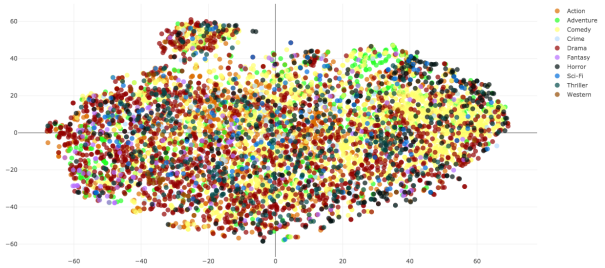

Figure 10: POS Tag Trigrams model, projection using Truncated SVD and t-SNE

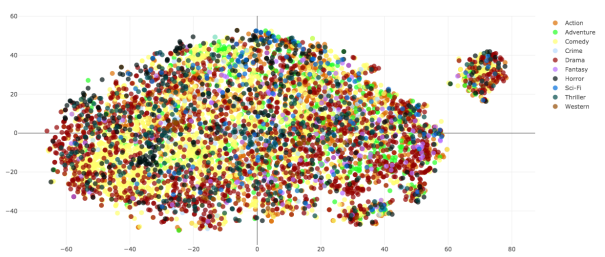

Figure 11: Stopwords Distribution model, projection using Truncated SVD and t-SNE

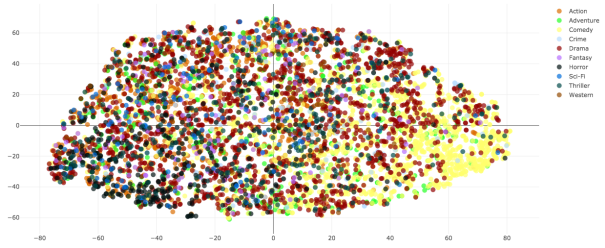

Figure 12: Stylometric & Statistical Measurements model, projection using t-SNE

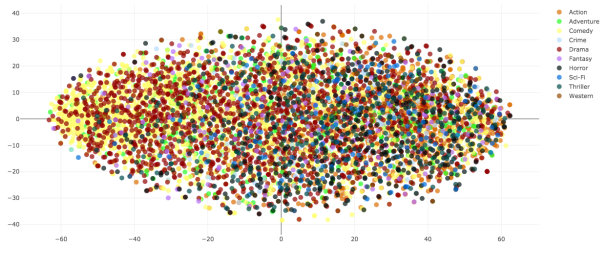

Figure 13: Sentiment Analysis model, projection using Truncated SVD and t-SNE

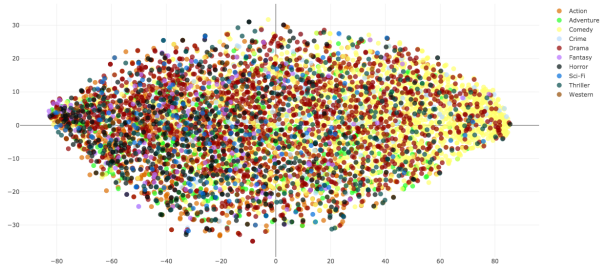

Figure 14: Speech Tempo Analysis model, projection using Truncated SVD and t-SNE

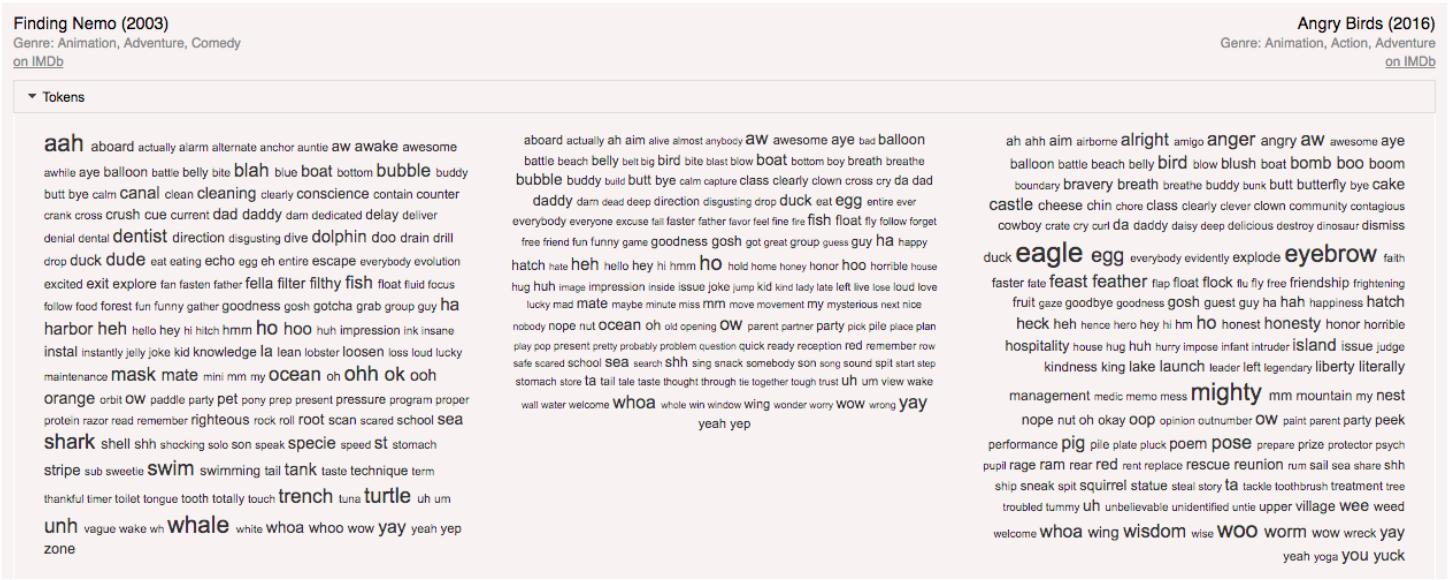

Figure 15: Detail view of Bag-of-Words model, showing most significant terms for ”Finding Nemo” (2003) and ”Angry Birds” (2016) and their intersection

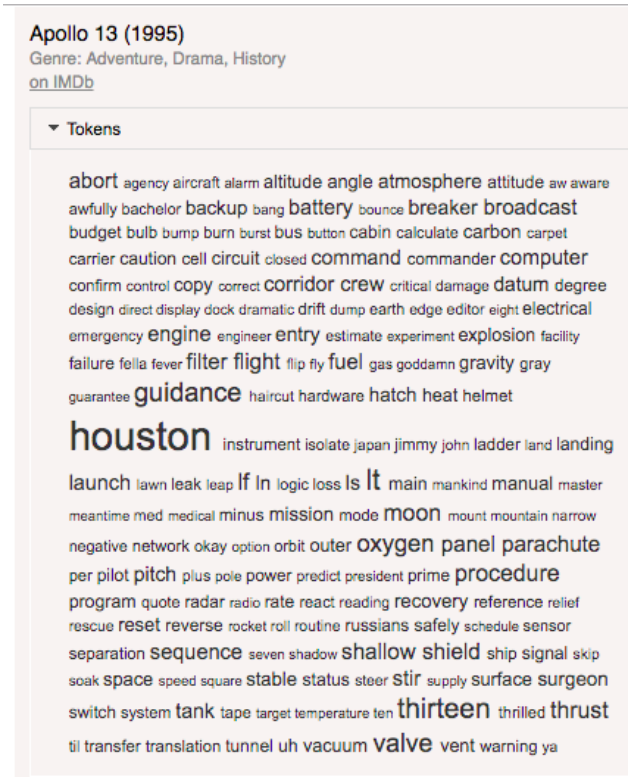

Figure 16: Detail view of Bag-of-Words model, showing OCR errors

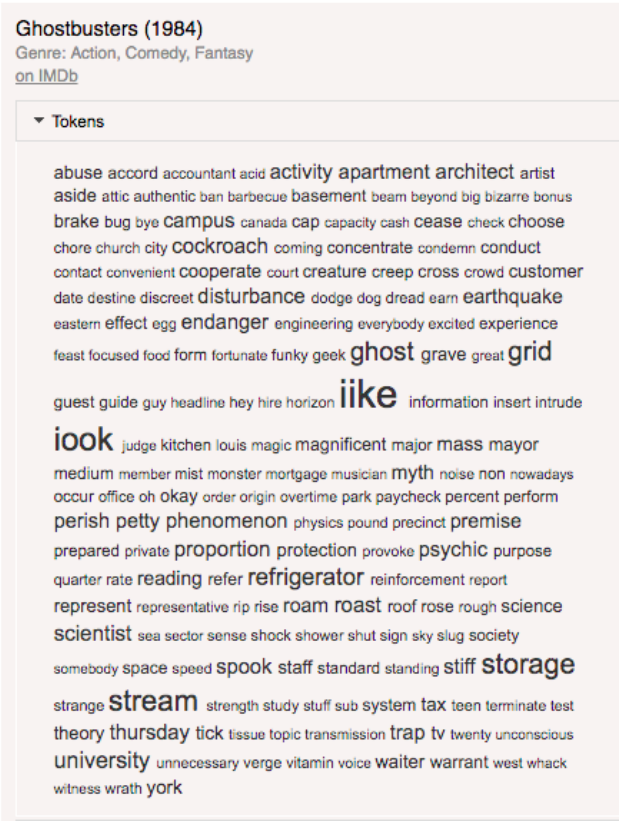

Figure 17: Detail view of Bag-of-Words model, showing OCR errors