Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Master Information Studies

Prediction Box Office Revenue & IMDb Rating Opening Weekend

Visual Analytics Final Report

ALEX OLIEMAN (6077285), STIJN VAN DEN BRINK (5922127), ROBIN SPIERINGS (10327053), MICHAEL WOLBERT (10277331)

Table of contents

Introduction

This report is written for the course Visual Analytics of the University of Amsterdam (UvA).
For this course we did a project based on the IEEE VAST mini-challenge 1. The goal of this project was to design and prototype a system that could facilitate the user in predicting the revenue on the opening weekend in the USA and the rating on the International Movie Database (IMDb[1]) of unreleased movies through visual analytics.

There are many others who have tried to predict movie success prior to their release. Oghina et al. (2012) attempted to predict IMDb movie ratings by mining Twitter, IMDb and YouTube data. The features they used can be divided in surface and textual features. Surface features are features that contain information *about* the tweet or post. The surface features they used are: number of tweets, comments, likes, dislikes, views, favourites, unique words / comments, and positive / negative term counts. Textual features are features that are pertaining to the actual content of the tweet or post. Textual features were used to find out if particular words can will be indicative of the movie's rating. They found that combining the like/dislike ratio from trailers on YouTube and the number of tweets mentioning the movie yielded the best results.
This is however quite different from our approach, since we performed this challenge as part of the UvA Visual Analytics course. The overall purpose of Visual Analytics according to North (2006) is to gain insight. He lists five characteristics of insight:

- ***Complex****:* Insight is a complex thing, it encompasses all, or large parts of the data in a synergistic manner. It's not just about isolated pieces of data.
- ***Deep****:* Insight is gained in a cumulative way over time, it raises new questions that inturn can evoke new insights. This is in agreement with the model by Keim et al. (2008) which integrated a feedback loop from Insight back to Source.
- ***Qualitative****:* Insight is far from exact and can be subjective, there can be multiple levels of resolution.
- ***Unexpected****:* Insight is often unexpected, coincidental and creative. This is why visualizing not only the obvious relations but also the less obvious ones can lead to interesting insights.
- ***Relevant****:* Insight is deeply rooted in the data domain. Data can get meaning by combining existing knowledge about the data with domain knowledge, it's more than just data analysis.

The data was selected by the VAST organizers and contained two weeks of Twitter data with tweets about the movie Texas Chainsaw Massacre 3D from the year 2013. The use of other data sources than the selected Twitter and IMDb data was prohibited. Based on North's definition of insight, we argue that the user would gain most insight if the data that is available will be combined and can be seen in context. This qualitative approach leaves room for the user to interpret the data and take his own experience and previous knowledge into consideration.

How the user gains insight is explained in the rest of the report. In the first part of the report the design of the system's front-end will be explained and underpinned, this includes decisions of color use, lay-out and how the various data available to us was used. It will be structured according to our interaction model since everything in the front-end system is included in this model, both the different states and interactions. Due to this structure the flow of the narrative gives a good idea of how the actual system works. In the second part of the report we will elaborate on the back-end of the system, what data has been used and how it was (pre)-processed to fit this project's needs.
Finally part three contains our evaluation on the system, the discussion and the possible improvements and additions that we see as opportunities for the future. It also includes an overview of the work distribution among the different authors.

[1] http://www.imdb.com/

Part 1 | Front-end System overview

The entire system for data analysts who want to predict the success of a movie based on Twitter and IMDb data.

1.1 | Movie prediction system overview

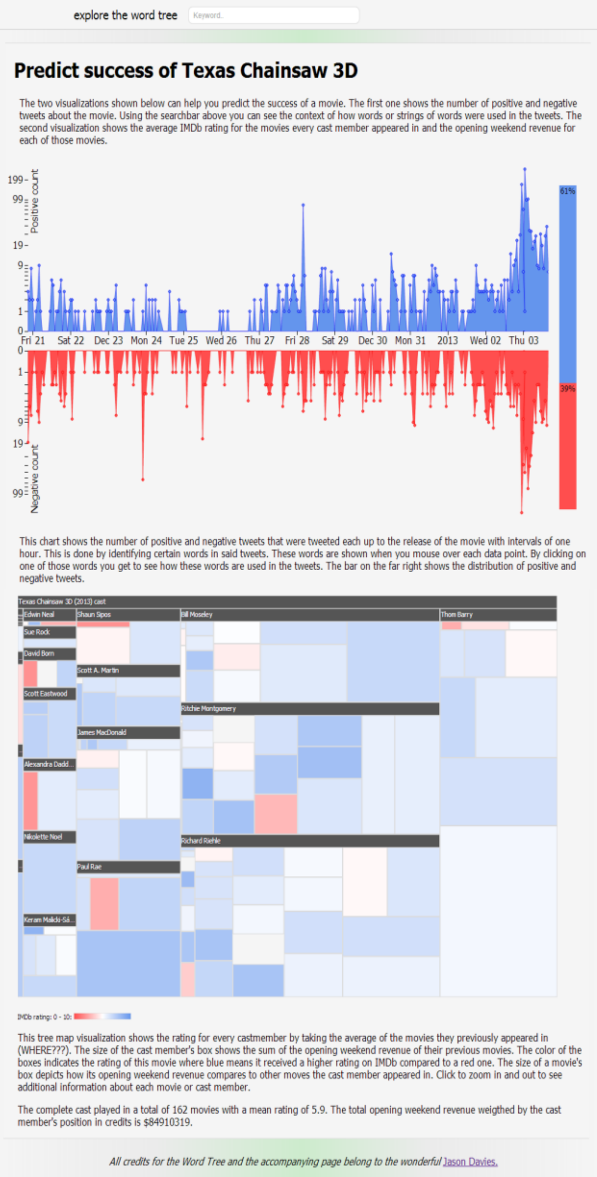
Narrative model

The visualization chosen is based on the partitioned poster genre of Segel and Heer (2010). This is the most appropriate form of visualizing because it contains multiple views. A multi-view visualization ("partitioned poster"), such as ours, may suggest only a loose order to its images. It is not as linear because the user can choose what view he would like to study and in what order. There is a little bit of directing since the wordtree is not visible at first for users. It can be made visible by accessing via the two other visualizations or the search field at the top of the webpage. Since the wordtree view is not visible at first sight, the second genre that can be identified is slide show, a path is identified: the two other visualizations are shown before the wordtree can be visible.

The structuring of the visualizations has been thought of as well. There are two equally important sub visualizations that we want to show on the screen simultaneously. Ideally we would place the two visualizations alongside each other. However the screen width is limited. Even though the tree map could be tilted this would not be preferred for the time line visualization. It would take more cognitive memory if tilted, and therefore it was decided to leave the visualizations in a horizontal orientation. We could not merge these visualizations because that would go against the rules of orthogonality.

Each individual visualization is more based on the annotated chart genre where one image is central and some additional information is given. On their own they are single views, however the magazine style does not apply here because would suggest that the visualizations are embedded in the text. In this case the text is supportive and not leading.

S1 Movie overview / Total visualization



Predict success of Texas Chainsaw 3D

The two visualizations shown below can help you predict the success of a movie. The first one shows the number of positive and negative tweets about the movie. Using the searchbar above you can see the context of how words or strings of words were used in the tweets. The second visualization shows the average IMDb rating for the movies every cast member appeared in and the opening weekend revenue for each of those movies.

This chart shows the number of positive and negative tweets that were tweeted each up to the release of the movie with intervals of one hour. This is done by identifying certain words in said tweets. These words are shown when you mouse over each data point. By clicking on one of those words you get to see how these words are used in the tweets. The bar on the far right shows the distribution of positive and negative tweets.

This tree map visualization shows the rating for every castmember by taking the average of the movies they previously appeared in (WHERE???). The size of the cast member's box shows the sum of the opening weekend revenue of their previous movies. The color of the boxes indicates the rating of this movie where blue means it received a higher rating on IMDb compared to a red one. The size of a movie's box depicts how its opening weekend revenue compares to other moves the cast member appeared in. Click to zoom in and out to see additional information about each movie or cast member.

The complete cast played in a total of 162 movies with a mean rating of 5.9. The total opening weekend revenue weigthed by the cast member's position in credits is $84910319.

All credits for the Word Tree and the accompanying page belong to the wonderful Jason Davies.

Figure 1.1: Overview of the opening page with both visualizations visible.

Interaction Design

Yi, Khang, Stasko and Jacko (2007) proposed seven general categories of interaction techniques widely used in Infovis: 1) Select, 2) Explore, 3) Reconfigure, 4) Encode, 5) Abstract/Elaborate, 6) Filter, and 7) Connect. These categories are organized around a user's intent while interacting with a system rather than the low-level interaction techniques provided by a system. We used them to identify what interactions the user can make with our movie prediction system. In figure 1.2 one can see the different states (S) and visualizations in the system along with the different types of interaction (I) that lead the user from one view to another.

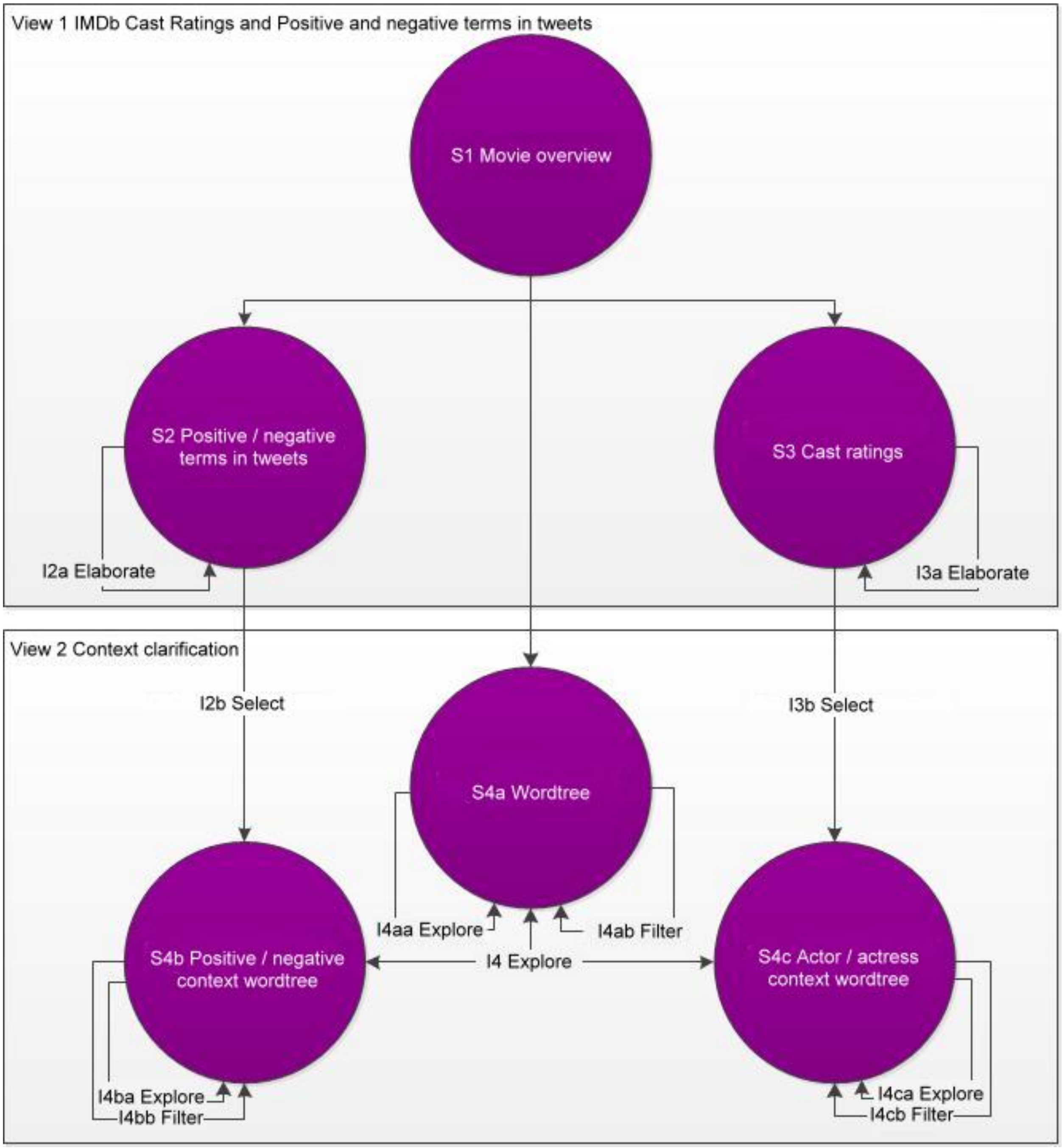

View 1 IMDb Cast Ratings and Positive and negative terms in tweets
S1 Movie overview
S2 Positive / negative terms in tweets
S3 Cast ratings
I2a Elaborate
I3a Elaborate
View 2 Context clarification
I2b Select
I3b Select
S4a Wordtree
I4aa Explore
I4ab Filter
I4 Explore
S4b Positive / negative context wordtree
S4c Actor / actress context wordtree
I4ba Explore
I4bb Filter
I4ca Explore
I4cb Filter

Figure 1.2: Interaction design model

1.2 | Individual states, visualizations and interactions

The entire system will be explained by going through the different states and interactions. for each of them will be explained why this view or interaction is chosen. For the visualization additional explanation is given how the use of, for example, color influences our perceptual and cognitive memory.

S2 Positive and negative terms in tweets

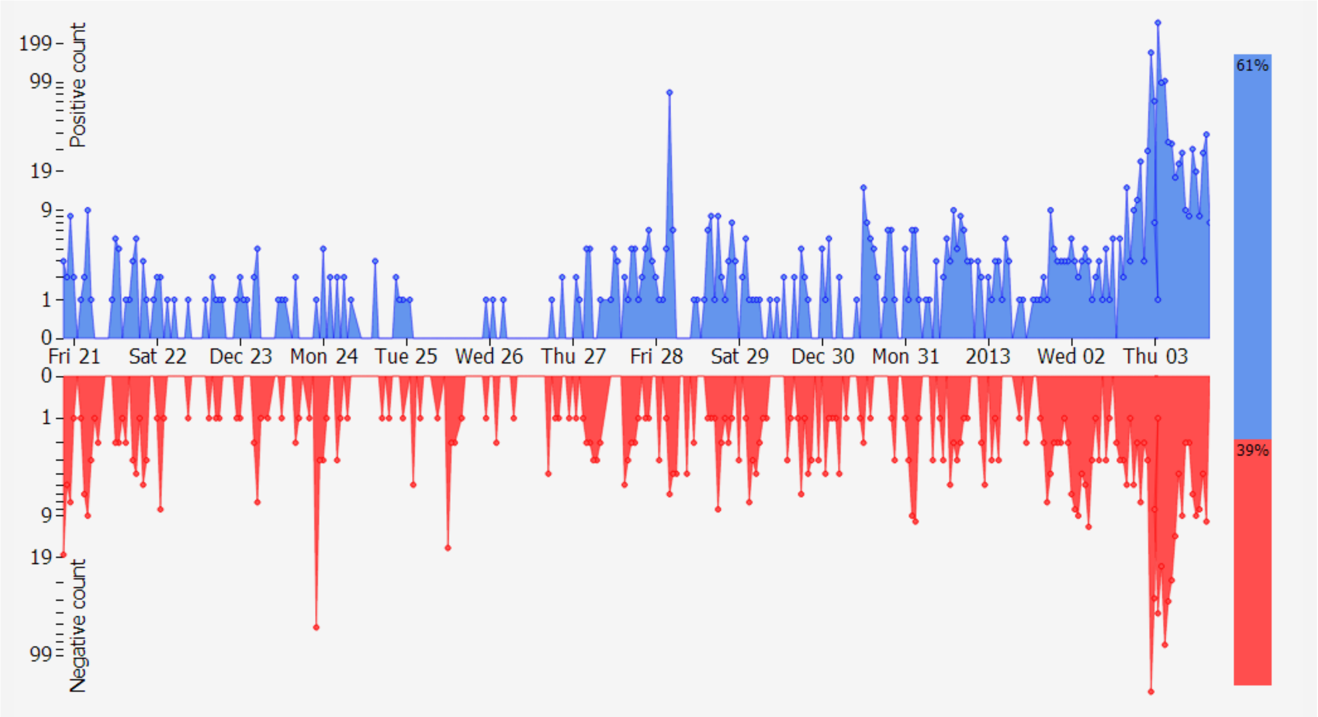


Figure 1.3: Chart visualizing the positive (blue) and negative (red) terms found in the tweets with on the right a bar visualizing the overall distribution.

Data	Type	Mapping
Positive term	Nominal	Tooltip on elements in positive graph
Negative term	Nominal	Tooltip on elements in negative graph
Positive term count	Ratio	Logarithmic y-axis in positive graph and its total distribution in bar chart (right)
Negative term count	Ratio	Logarithmic y-axis in negative graph and its total distribution in bar chart (right)
Tweet creation time	Ratio	X-axis of both positive and negative graph.

Table 1.1: Data to visualization mapping for the tweet chart.

The visualization shows the graph with the number of recognized positive and negative terms in tweets right up until the movie is released. Oghina et al. (2012) have shown that certain terms in tweets are good indicators of the IMDb rating that a movie will receive after its release. As such, these term occurrences are an important success indicator for movies and they take a prominent place in our approach to visual movie success prediction.

This graph combines features from an area and a line chart to make it easier to read individual values while still allowing for the user to see differences between the number of positive and negative term occurrences. The logarithmic scales do make this a little harder, but they had to be used in this case. A linear scale made it extraordinarily hard to see the difference between low values, for example between 1 and 5. So although this makes it less intuitive to see the ratio between individual values, it does create a clearer graph near the x-axis. Another measure that was implemented to reduce clutter was the removal of circles for hours when no positive or negative words were mentioned in any of the tweets. This also improved readability drastically.

The decision to use the colors blue (for positive) and red (for negative) instead of, the perhaps more obvious, green and red was made because of the high prevalence of red-green color blindness. The bar on the right shows how the total number of tweets are distributed. We argue this is a necessary addition, because the main graph employs logarithmic axes, which can make it difficult to spot absolute differences between individual values.

Elmqvist and Fekete (2010) define six guidelines for visualizing aggregates, which we discuss for our visualization:

1. Entity budget: Predefine the max number of entities you allow on the screen.
2. Visual summary: Aggregates should convey information about the underlying data.
3. Visual simplicity: Aggregates should have a clear and simple visual appearance.
4. Discriminability: Aggregates should be visually distinguishable from individual data points.
5. Fidelity: accuracy of the visual aggregate.
6. Interpretability: Aggregate items only so much that the aggregation is still correctly interpretable within visual mapping.

Even though no entity budget was taken into account when creating this visualization, there is little to no clutter. This visualization does conform to the second guideline in that it counts the number of tweets that were either identified as being positive or negative. Also, because a ‘count’ is an intuitive concept which we visualize as simple points, we argue that the visualization complies with the third guideline. The fourth guideline is not applicable because no individual data points are shown. The fifth guideline is breached, since the used logarithmic scale (y-axis) could lead to an inaccurate comparisons of values of the visual elements. As for the last guideline, interpretability, this largely depends on the size of the dataset. For this dataset aggregating tweets for every hour worked out great, but when you use months worth of tweets, aggregating every hour might give the user too much detail.

Another set of guidelines that should be taken into account when building visualizations are the Gestalt principles (Graham, 2008). These principles describe how humans perceive objects as belonging to the same group. We identified seven principles:

1. Proximity: this law states that if everything else is equal, objects that are placed near each other are perceived as belonging to the same group.
2. Similarity: if all else is equal, humans perceive objects as belonging together when they resemble each other.
3. Closure: this rule says that humans are predispositioned to see complete figures, even if there are gaps or the object is partially hidden.
4. Continuity: humans have the tendency to group lines that follow an established direction instead of those that have sharp changes in direction.

5. Symmetry: useful for showing similarities and differences.
6. Relative size: smaller components tend to be seen as objects.
7. Figure and ground: a combination of symmetry, whitespace and closed contours can cause the object to be perceived as a figure.

Another very strong stimulus, but not part of the Gestalt principles is connectedness: this law says that even though two objects are similar or near each other, if they are connected to another object the connected ones will be perceived as belonging together.

This visualization employs the symmetry law. By putting the graphs on opposite sides of the x-axis, seeing similarities and asymmetries becomes much easier compared to when they would have been stacked on top of each other in a stacked chart.

I2a Elaborate

Using a tooltip this visualization provides the user with the possibility to see what positive or negative terms were used in tweets each hour. This gives the user extra information and is thus an elaborative interaction.

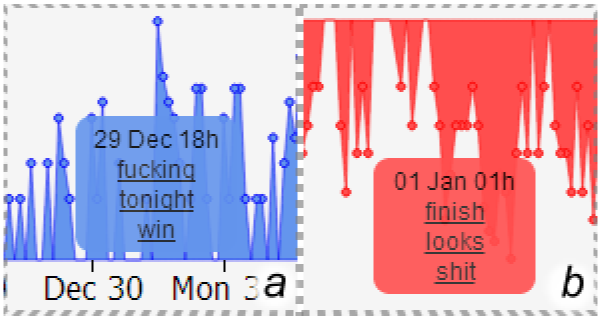


Fig 1.4: The tooltips shown when the user hovers over a datapoint. A) positive terms. B) negative terms.

In S2 Positive / negative ratio of terms and tweets the user can hover over the different points in the graph. When the tooltip crosses one of the data points in the graph the user gets details on demand, as mentioned in the Shneiderman Mantra (Shneiderman, 1996), to prevent information overload. This form of elaborating provides the user with the ability to adjust the level of abstraction of a data representation. This is helpful because different levels of abstraction are suited for different stages in the interaction. At first the user would probably want an overview before she wants to view the details for remarkable points in the graph.

I2b Select

If the user finds something remarkable and would like to see the context of the negative or positive terms that are used he can click on the terms and then gets redirected to S4b Positive / negative context wordtree. The root word will be the word that the user clicked on in S2. Since he clicks on an item, either the positive or negative term, to keep track of it, this can be identified as select.

S3 Cast Treemap

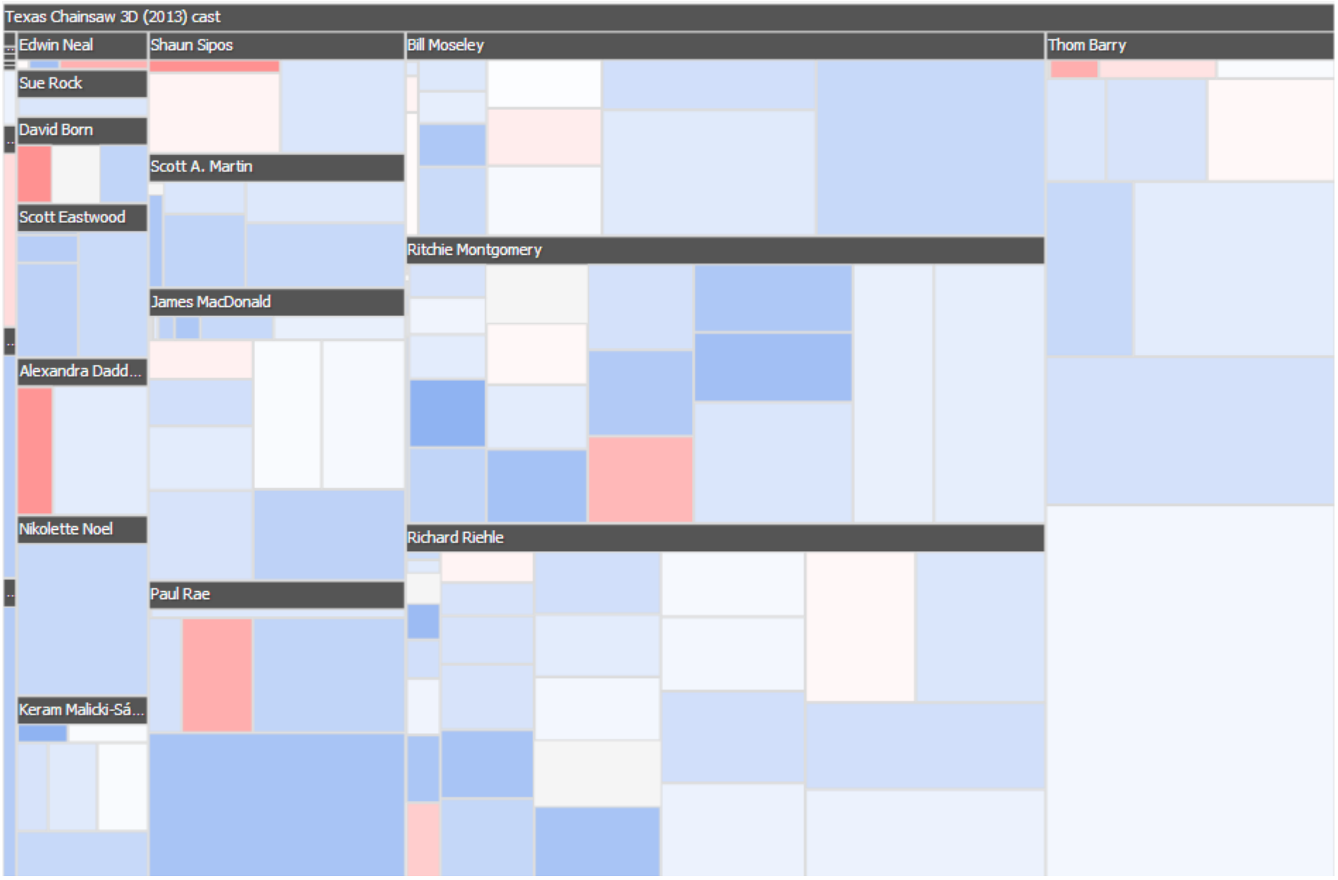


Figure 1.5 | S3 Cast Treemap: The cast played in a total of 162 movies with a mean rating of 5.9. The total opening weekend revenue weighted by the cast member's position in credits is $84,910,319.

Shneiderman (1991) introduced the treemap (see fig 1.5), a 2-d visualization for tree structures, where space filling algorithms are employed to visualize its nodes. Each leaf node should at least have a quantitative coded variable for the space filling algorithm to determine the relative size of the element for visualization. Furthermore, Shneiderman argued that for visual clarity, color coding should be used to further aid the user in making sense of the underlying data.

We hypothesized with the historical IMDb cast and movie data that performance of the cast in movies prior to the current movie indicates its success in terms of opening weekend box office and movie rating. In this case, the cast’s performance is expressed by movie rating and its opening weekend box office. We assumed that the credit position of each cast member could be taken as a weighting factor of the opening weekend box office. In this case, cast members with a lead role (i.e. 1st credit position) contributed more to the opening weekend box office than support cast members. As stated by Shneiderman (1991) the use of color, in our case a bi-polar range (red - white - blue), was chosen to communicate the rating of the movies on a scale from 0 to 10.

Data	Type	Mapping
Cast performance current movie	Tree	Root node and subsequent nodes of treemap
Cast member’s name	Nominal	Parent rectangle text
Cast member’s opening weekend	Ratio	Parent rectangle size
Movie name	Nominal	Child rectangle text
Movie opening weekend	Ratio	Child rectangle size and text on details
Movie rating	Ratio	Child rectangle color and text on details

Table 1.2: Data to visualization mapping for the Cast Treemap

Initially, we preprocessed the cast performance data from four movies (see Figure 1.6) and created a treemap in Tableau by aggregating on cast member. Size and color visual elements were set to opening weekend box office and movie rating respectively. Note that the movie rating for each cast member is the average of all movies in the data for the given cast member. From these visualizations the hypothesis on cast performance can be accepted; for instance, comparing treemaps for one of the bottom rated movie on IMDb (Disaster Movie) with one of the top rated movies (Pulp Fiction) shows that the cast of the latter movie is ranked much higher. Moreover, we gained the insight that the movie Texas Chainsaw Massacre 3D will have a mediocre rating (e.g. 5.0 - 6.0). While contribution of each cast member to prior opening weekend box offices is clearly depicted in the treemap, its prediction for the current movie, however, poses an issue. Since all treemaps share similar size but differ on total amount of opening weekend box office, these visualizations can not be compared. Therefore we devised a textual description (in a poster design fashion) along with the treemap to visualize the measurement directly (see caption of Figure 1.5).

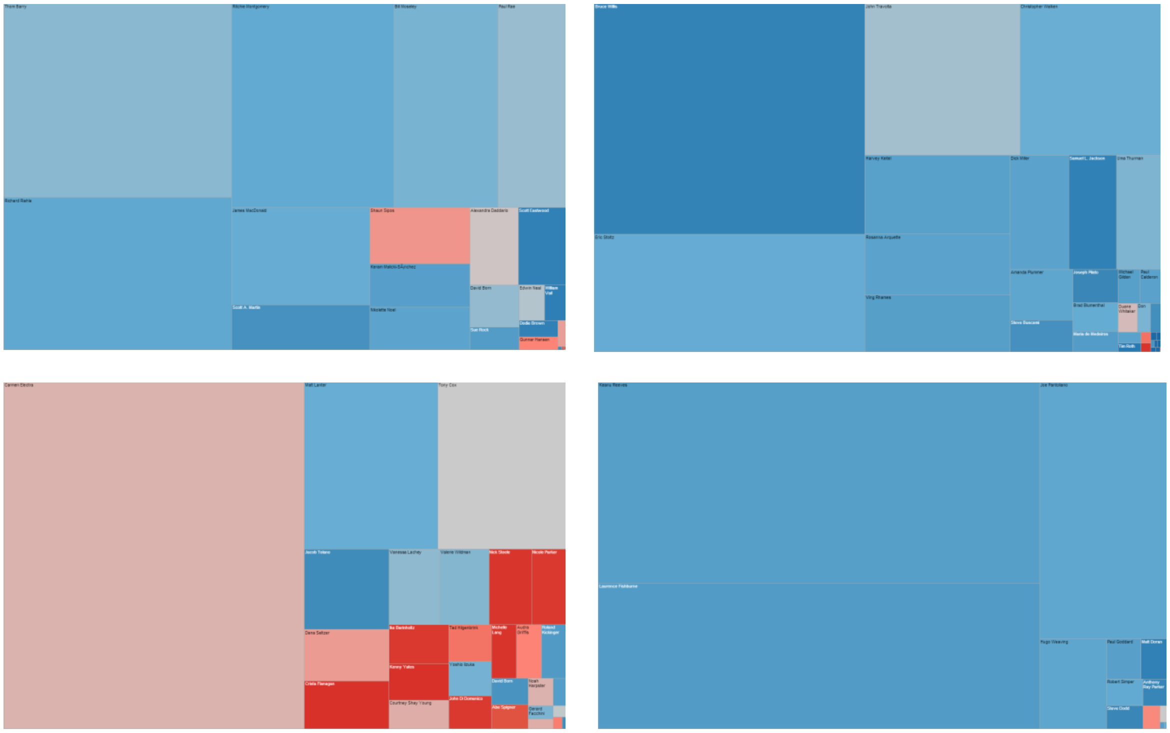


Figure 1.6 **|** *S3 Intermediate Cast Treemaps: from the upper left treemap in clockwise direction the cast for movies:* ***Texas Chainsaw Massacre 3D*** *(2013),* ***Pulp Fiction*** *(1994, IMDb rating: 9.0, IMDb opening weekend box office: $9,311,882),* ***Disaster Movie*** *(2008, IMDb rating: 1.9, IMDb opening weekend box office: $6,945,535),* ***The Matrix*** *(1999, IMDb rating: 8.7, IMDb opening weekend box office: $27,788,331).*

Since the insight was established that prior cast performance proves to be a good indicator for movie rating, we enhanced the design by visualizing more information of the underlying aggregation and adding interaction (see I3a Elaborate). In the final treemap we employed squarified space filling to maintain good aspect ratios of the rectangles for each cast member (von Landesbergen et al., 2011). These parent rectangle sizes are based on an aggregation with a sum function. Here, we tried to follow the design principles for aggregation (Elmqvist & Fekete, 2010). We discuss these below and indicate how these relate to Gestalt principles (Graham, 2008).

Entity budget was not considered, whereas all data is visualized. This may result in visual clutter, especially when cast members have a supportive roles in a movie (resulting in a very small sized visual element). On the other hand, when applying *entity budget*, crucial information may not be visualized, resulting in less accurate movie success predictions. *Visual summary* is achieved by grouping visual elements (movies) for each cast member based on *proximity*; visual elements which represent movies are visualized in a parent element (*closure*) which represents the cast member. Similarly, child elements exhibit *closure*. *Visual simplicity* is achieved by conveying the most important information for processing on the low- level feature detectors of our brain (*color*, *relative size*). Furthermore, the space filling algorithm of the treemap proposes *continuity* since it places visual elements in such a way that aspect ratios are maintained without resulting in a garbled presentation of elements. *Discriminability* is achieved by visualizing parent (cast member) and children (movies) differently: the parent header is given a color not available to the child elements. With the intermediate cast treemaps *interpretability* was not achieved since the visual mapping lost information regarding each individual movie. Moreover, the average function of the aggregate posed this issue. Therefore, we decided to visualize each movie of a cast member as the smallest element in the visualization to reflect the actual visual mapping in a correct manner.

I3a Elaborate

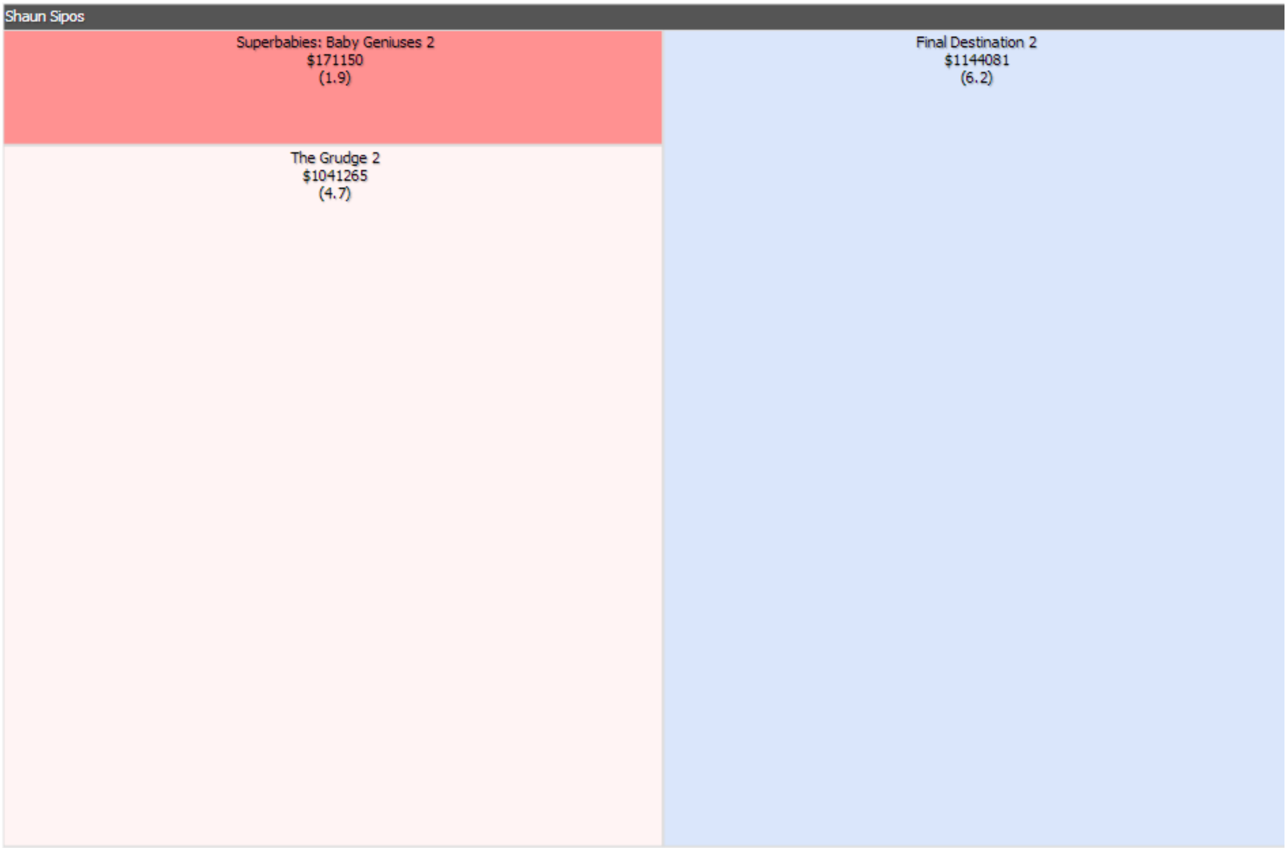


Figure 1.7 **|** *S3 Cast Treemap and the information for a cast member once zoomed in.*

This interaction refers to the above mentioned zoom function in the tree map. By zooming in the user gets to see additional information about the movie the cast member was in. The visualization now shows the title, IMDb rating and the opening weekend revenue divided by the cast member's position in the credits for every movie this cast member was in.

I3b Select

Going from the treemap of the cast S3 the user can select an interesting name of one of the actors or actresses and see if this name is mentioned in one of the tweets in the dataset. The user can click the name of the actor or actress and then proceeds to view the wordtree. The name of the actor or actress will, in this case, be the root node of the wordtree.

S4a Wordtree

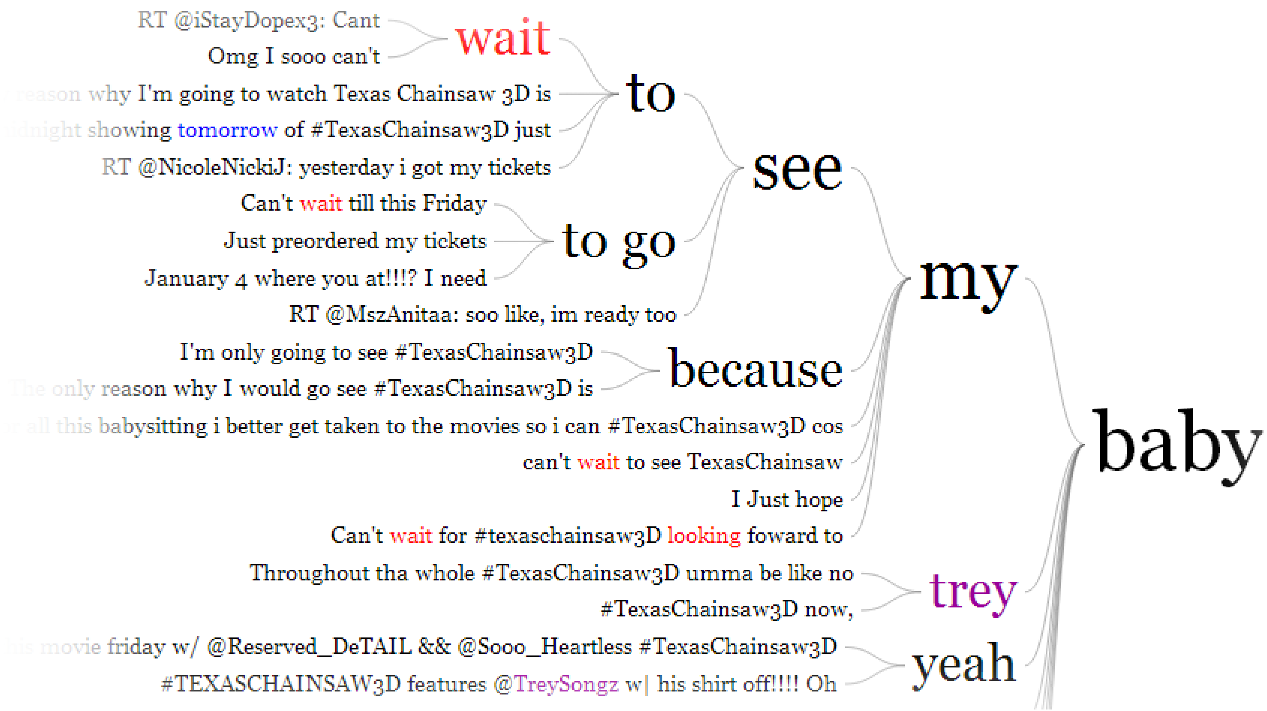

RT @iStayDopex3: Cant
Omg I sooo can't
wait
reason why I'm going to watch Texas Chainsaw 3D is
midnight showing tomorrow of #TexasChainsaw3D just
RT @NicoleNickiJ: yesterday i got my tickets
to
Can't wait till this Friday
Just preordered my tickets
January 4 where you at!!!? I need
to go
RT @MszAnitaa: soo like, im ready too
see
I'm only going to see #TexasChainsaw3D
The only reason why I would go see #TexasChainsaw3D is
because
y all this babysitting i better get taken to the movies so i can #TexasChainsaw3D cos
can't wait to see TexasChainsaw
I Just hope
Can't wait for #texaschainsaw3D looking foward to
my
Throughout tha whole #TexasChainsaw3D umma be like no
#TexasChainsaw3D now,
trey
this movie friday w/ @Reserved_DeTAIL && @Sooo_Heartless #TexasChainsaw3D
#TEXASCHAINSAW3D features @TreySongz w| his shirt off!!!! Oh
yeah
baby

Figure 1.8: The wordtree showing phrases and words used in tweets that contained the word 'baby'.

Data	Type	Mapping
Tweet text	Nominal	Tree structure
Positive word	Nominal	Font color (blue)
Negative word	Nominal	Font color (red)
Cast member	Nominal	Font color (purple)
Word count	Ratio	Font size

Table 1.3: Data to visualization mapping for the wordtree.

There is a lot of information concealed in the text of the tweets that is not disclosed by any of the previous visualizations. The Word Tree is presented by the original designers as "a new visualization and information-retrieval technique aimed at text documents" (Wattenberg, & Viégas, 2008, pp. 1221), and was developed for use in IBM's ManyEyes. This technique originates in concordances or, more recently, "keyword in context" methods, that produce an overview of the contexts in which a certain term in a text is used.

Whereas previous methods fill the available space by vertically listing sentences in which the keyword is centered horizontally, the word tree combines multiple sentences into a tree structure in which the keyword is the root node. It maps the term frequency of subsequent terms onto the font size, which allows frequently occurring phrases to be spotted immediately. The most important interactivity is the ability of the user to expand the root node into a phrase by clicking on subsequent terms. This narrows down (i.e.

filters) the result set. It is also possible to move the focus of the word tree onto another word (i.e. explore) with an alternative click command. Finally, the tree can be reversed to view preceding terms instead of subsequent terms.

A possible use for the word tree is to investigate the context of tweets in which a certain actor is mentioned (see S4c), and to view the context in which recognized positive and negative terms occur (see S4b). This visualization provides for these use cases by allowing a user to select a cast member's name or a positive/negative term (I2b & I3b), and use the selected word(s) directly as root node in the word tree. The functionality of the word tree is, however, not limited to this envisioned use, and can also be employed to explore the content of the tweets freely. To aid this, we show all tweets in a sidebar, in chronological order. An interactive scrollbar that separates the sidebar from the word tree also serves to show all occurrences of the root node, thereby providing additional context of the selected term in the timeline of all available tweets.

This visualization draws on the connectedness and size Gestalt principles (Graham, 2008). Connectedness is used to indicate that certain words or phrases were used in a certain order, and the size indicates how often this happened. The more often it occurred, the larger the font so that phrases that were often uttered on Twitter draw the user's attention. The word tree, as a directed graph, also preserves the word order of the original text. Instead of expressing the direction of connections with arrows, the left-right reading direction is exploited by the layout algorithm so the wordtree can be read naturally.

Finally, we discuss how well the wordtree complies with the guidelines for aggregation (Elmqvist & Fekete, 2010). *Entity budget* is considered by not drawing any subtrees which would take less than 3 pixels of vertical space. Instead, only the deepest branch in the subtree is drawn (Wattenberg, & Viégas, 2008). Aggregates are *visual summaries* of the underlying data by showing the term with multiple occurrences as a separate node with a font size that is proportional to the square root of the number of occurrences (ibid.). *Visual simplicity* is maintained by drawing the term only, and by omitting additional information such as author or creation time. The *discriminability* between tweets and aggregates relies upon drawing the aggregates as separate nodes, with an increased font size. However, by making the font size of aggregates proportional to the square root of the frequency of occurrence the *fidelity* is somewhat affected, because the visual ratio differs from the ratio of frequencies in the actual data. Finally, the *interpretability* is preserved by using whole phrases as leaf nodes, instead of single terms.

S4b Positive / negative context wordtree

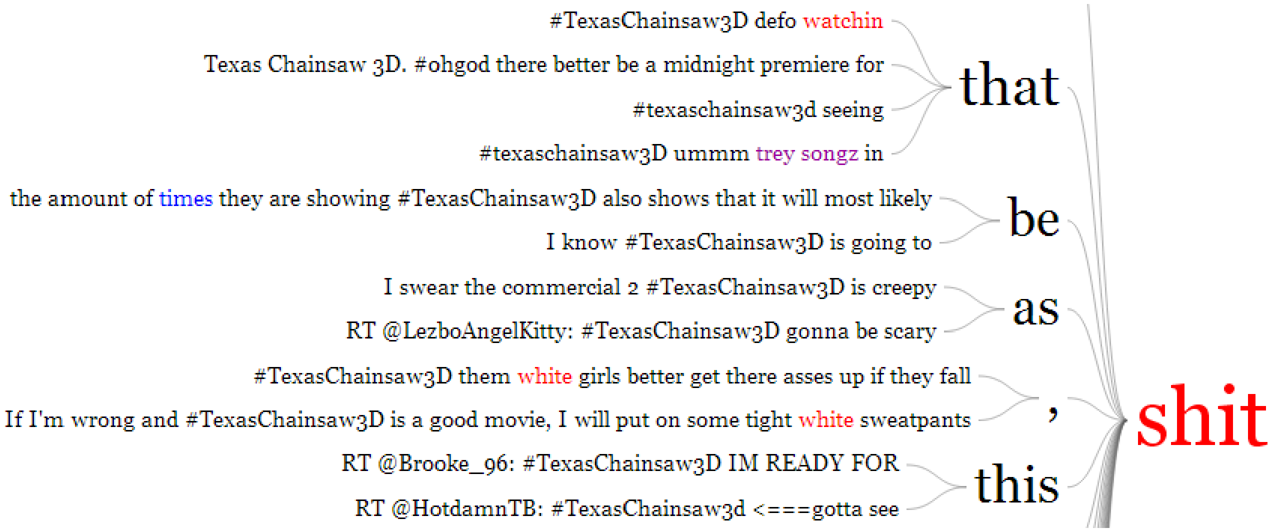

#TexasChainsaw3D defo watchin
Texas Chainsaw 3D. #ohgod there better be a midnight premiere for
#texaschainsaw3d seeing
#texaschainsaw3D ummm trey songz in
that
the amount of times they are showing #TexasChainsaw3D also shows that it will most likely
I know #TexasChainsaw3D is going to
be
I swear the commercial 2 #TexasChainsaw3D is creepy
RT @LezboAngelKitty: #TexasChainsaw3D gonna be scary
as
#TexasChainsaw3D them white girls better get there asses up if they fall
If I'm wrong and #TexasChainsaw3D is a good movie, I will put on some tight white sweatpants
,
RT @Brooke_96: #TexasChainsaw3D IM READY FOR
RT @HotdamnTB: #TexasChainsaw3d <===gotta see
this
shit

Figure 1.9: Wordtree showing highlighted positive (blue) and negative (red) words found in tweets.

The content of these tweets is used to create a word tree wherein the user can see the context of how positive and negative words are used. We propose this is a useful feature because there is often some ambiguity in the use of terms that are recognized as positive or negative, such as: “gotta see this shit” (see Figure 1.9). The term “shit” is a clear indicator for a negative IMDb rating, but when someone tweets “gotta see this shit”, it likely also means that he or she did at least plan on watching the movie, thus potentially contributing to a higher opening weekend revenue. This word tree allows the users to see in what context these words were used so they don’t have to completely rely on the graph which can contain ambiguities as shown in the above example.

S4c Actor context wordtree

The wordtree is used to view the context of the tweets in which one of the cast members is mentioned by name. This functionality can be used to estimate how much of an audience will be drawn to the movie by the presence of a certain actor or actress. As such, this functionality complements S3 Cast Treemap, by making it possible to gain similar insight from an independent dataset. This can be especially useful for cast members who haven’t starred in many previous movies, in which case the IMDb data arguably gives a less accurate picture than the Twitter data.

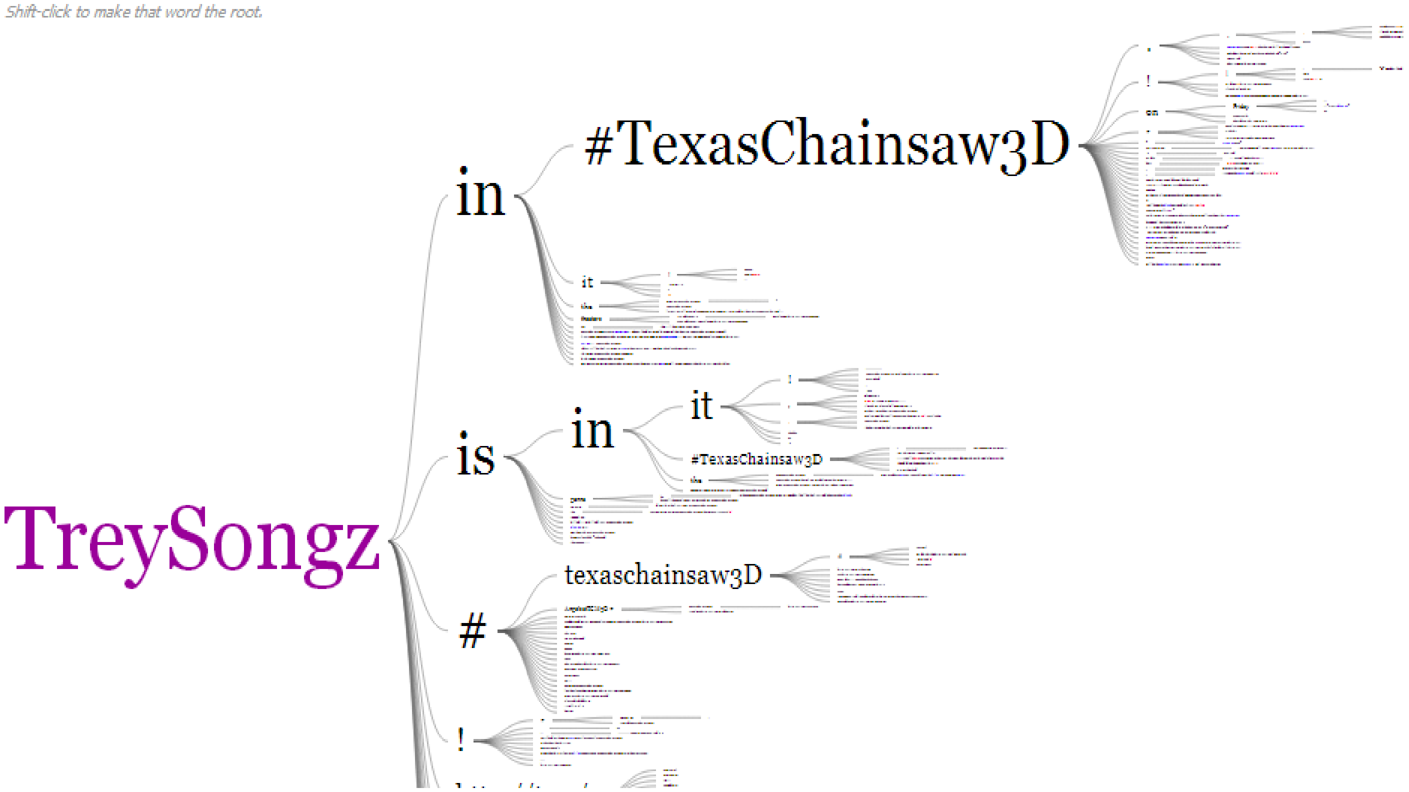


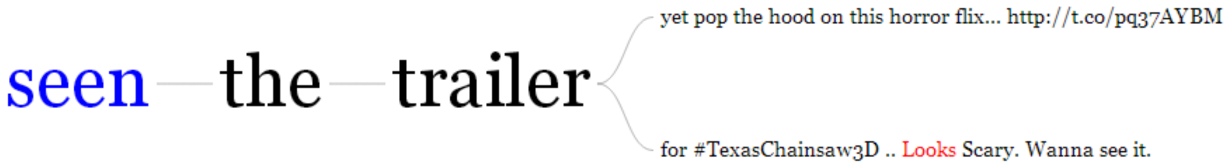
Figure 1.10: Wordtree showing how cast member mentions found in the tweets are highlighted in purple.

I4a-ca Explore

At all times the user is able to set a new root node for the wordtree by using the search field or an alternative click command. This interaction behavior leads to the user's ability to examine a different subset of tweets. Subsequently, filtering (see I4ab) is used to focus on a particular phrase and limit the number of data items shown.

I4a-cb Filter

Once the root node of the wordtree has been set (see I4aa), the user is able to click additional terms to create a phrase. This interaction behavior leads to visualizing tweets conditionally based on phrase occurrence (see Figure 1.11). Moreover, this interaction filters the initial subset of tweets, and allows the user to gain further insight in what context the phrase has been tweeted. This filter operation is animated, smoothly fading out the tweets that do not match the filter, and moving the remaining tweets to their new location.



yet pop the hood on this horror flix... http://t.co/pq37AYBM

for #TexasChainsaw3D .. Looks Scary. Wanna see it.

Figure 1.11: Wordtree that only shows tweets that used the phrase "seen the trailer".

Part 2 | Back-end system overview

This section describes which data was used and in what way it was used. For a complete overview of the available data please refer to appendix A. In this section we mention the used data sources and elaborate on how they are (pre-)processed. The source code of the system is hosted and publicly available on GitHub[2].

IMDb

In the cast treemap (S3) section we discussed what IMDb data was used to visualize the treemap. The IMDb text interface[3] are used as the source. The following text files were retrieved: 1) actors.list, 2) actresses.list, 3) movies.list, 4) ratings.list and 5) business.list (includes opening weekend box office). Since the data from this source could not be used directly, preprocessing was needed. We used IMDbPY[4] to transform the text files to a SQL database which made querying the data more effective through the module's infrastructure. A python script which takes a movie title as input and queries the database for previous movies (movie rating, opening weekend box office) was employed to process the data. Minor sanitization has been done to parse opening weekend box office from a string (ie. '$1,000,000') and divide this value by the credit position of the given cast member. The resulting output (see code 2.1) is a tree structure formatted in JavaScript Object Notation (JSON).

```
{
        name: "Texas Chainsaw 3D (2013) cast",
        children: [{
                name: "Alexandra Daddario",
                children: [{
                        name: "Hall Pass"
                        rating: 5.9,
                        opening_weekend: 1127947,
                },
                {
                        name: "Jonas Brothers: The 3D Concert Experience"
                        rating: 2,
                        opening_weekend: 431392,
                }]
        }]
}
```

Code 2.1: JSON output of python preprocessing script. The root node of the tree structure represents the current movie, its children represent cast members and their children represent previous movies in which the given cast member had a part.

Twitter

The graph of positive and negative terms in tweets (see S2) and the word tree (see S4) are both driven by a dataset of tweets that mention the movie under investigation. Of all the attributes of the Twitter data, the timestamp at which the tweet was created, and the textual content were found sufficient to implement

[2] https://github.com/mchlbrnd/visual-analytics-2013-boxoffice

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

[4] http://imdbpy.sourceforge.net/

the designed visualizations. Extensive preprocessing was, however, necessary to arrive at the visualizations.

The main goal of this processing was to recognize occurrences of the positive and negative terms that have been found to be strong indicators for high and low IMDb ratings by Oghina et al. (2012). Initially, a python script was created which reads each tweet from a comma-separated table, applies a punctuation tokenizer (from the NLTK[5] module) to separate words from the tweet string, and employs a snowball stemmer (also NLTK) to compare each word to the sets of stemmed terms by Oghina et al. (2012).

The script hence expands the table with three variables: the count of positive term occurrences, the count of negative term occurrences, and a version of the tweet in which positive and negative terms are annotated with HTML classes. The resulting table was used to create preliminary visualizations in Tableau and HTML. These visualizations were highly cluttered and led to the observations that aggregation could be used to create an overview of positive and negative term occurrences, and that filtering could be employed to view the textual content of the tweets.

A second python script was written to produce temporal aggregations of the positive and negative term occurrences. Because the timespan of the Twitter data comprised several weeks, we found that an hourly resolution would convey sufficient information about the underlying data. The script parses the creation timestamp of each tweets and sums the positive and negative term occurrences for each tweet that was created in the same hour. Additionally, the unique terms that occur in that hour (i.e. duplicates are removed) are saved as alphabetically sorted lists. The resulting output (see Code 2.2) is a list in JSON format.

```
{
        "hour": "2012-12-28 15h",
        "neg_count": 1,
        "neg_terms": [
                "wait"
        ],
        "pos_count": 6,
        "pos_terms": [
                "awesome",
                "win"
        ]
}
```

Code 2.2: JSON output of python aggregation script for one hour. The entire output is a list of such “hourly sum” objects.

Web application

The partitioned poster has been implemented on a single HTML page which is hosted on a HTTP server. The data, discussed in the sections above, is hosted in a similar fashion. Each visualization is implemented in separate JavaScript files, which in turn are included on the single page. Since HTML was used to present the system, the decision was made to use D3.js[6] to map the data to visualizations and to support the designed interactions.

[5] Natural Language Toolkit - http://nltk.org/ - Bird, Loper, & Klein, 2009
[6] D3.js (Data Driven Documents) - http://d3js.org/

To draw the wordtree, we made use of an existing implementation[7] in JavaScript and D3.js. This implementation is initialized by us with a plain text version of the tweet contents, in chronological order. Also, we have changed the default parameters of the implementation to better match our dataset (e.g. to draw one phrase per line). Furthermore, we expanded the functionality of this implementation to draw positive and negative terms, and names of cast members in different colors. To do this, we edited the function which draws terms as SVG elements. Each term is stemmed with a snowball stemmer[8], after which it is compared to the terms in three separate lists for positive terms, negative terms, and cast member names. If the stemmed term is found in one of the lists it is colored correspondingly (see Table 1.3).

Part 3 | Reflection

Evaluation

At the start of this paper we described insight as being complex, deep, qualitative, unexpected and relevant. We think the above described system will aid the user in getting that insight by providing them with the right tools. The tool was not intended to provide an accurate prediction in terms of a quantified movie rating and box office success. Since its use results in a qualitative prediction, this arguably leads to a tool that is very hard to evaluate. Whether this tool provides someone with insight depends on the data that is used and who is using it. If the data does not contain anything interesting even the best visualizations will not give anyone any insight. And because of the limited time available for this project, and lack of access to the intended users, none of the existing evaluation techniques could be employed to see how effective this system actually is when it comes to facilitating user insight.

Discussion / Future work

Due to the fact that this project was subject of time limitations and rules, there are certain features and ideas that did not make it to the final product. First and foremost the initial idea was to have the user choose what movie was used for the visualizations, but due to the available data and time the decision was made to chose a specific movie for the visualizations. If this were to be implemented the user would be provided with a new start screen in which he can choose a movie. This would result in a new state for the interaction model in which the user can filter the data to only use relevant data for the visualizations.
For the graph visualizing tweets a word cloud showing used words in tweets for a certain time period decided by the user could give the user a nice idea how people feel about the movie changes over time.
The way cast members are recognized in the tweets currently is by using a small table that contains aliases and Twitter handles. The IMDb data also contains a list of aliases for every actor and adding this list to the table we now use would be a great addition because people can use a whole range of different ways to call an actor. By also using this list the system would be able to highlight more of these different ways. Besides the list of aliases, the IMDb data also contains the part every cast member played. By also using this list the system could highlight these names and give an even better view of how people feel about the cast members.
The treemap present in this tool is based on an example of a hierarchical treemap. An important feature of this example was that it supported zooming on parent nodes for elaboration of the underlying data. In our case the underlying values of the visual elements were visualized directly. Ideally, the cast member's rectangle would be given a uniform color in the overview state, showing the average rating for every

[7] JavaScript wordtree by Jason Davies - http://www.jasondavies.com/wordtree/
[8] jsSnowball - https://github.com/fortnightlabs/snowball-js

movie. This overview would have a similar appearance to the intermediate treemaps (see Figure 1.6). The individual movies, their rating and the opening weekend revenue would only be shown when zoomed in. This arguably makes it easier to quickly get an overview of prior cast performance, while preserving the possibility to derive more details when zooming in.

In addition to the above it would be desirable for an even more nuanced and complete prediction to have more data available from different sources. In this case we were bound by the limitations as given by the IEEE VAST challenge. However, in the future more data sources, such as IMDb's cast ratings, geographically tagged tweets, or YouTube comments, likes, and dislikes on movie trailers would be of great added value. According to Orghina et al. (2012) the ratio between likes and dislikes from YouTube videos would provide a reliable indication for movie success. In combination with the information and visualizations that are already provided in our system this would lead to more insight for the user according to North's definition of insight as mentioned in the introduction.

Team

Name	**Responsible for**
Alex Olieman	**Data and implementation** - Twitter data processing - Preliminary Twitter visualizations - Website implementation - Positive / negative term timeline in D3 - Expanded wordtree functionality **Report** - Editing figures - Wordtree - Twitter data processing - Web application
Michael Wolbert	**Data and implementation** - Tweet fetcher with caching (unused) - Initial Twitter data processing - IMDb data processing - Tableau intermediate treemaps - Website implementation - Treemap D3 implementation - Tweet positive/negative term barchart in D3 **Report** - Editing figures - Cast treemap - IMDb data - Web application
Robin Spierings	**Presentation design** - Presentation document + underpinning - Narrative structure / design of presentation - Present the presentation **Report** - Front page design - Introduction - Narrative design system - Interaction model design + drawing. - Interaction explanations of design decisions

Stijn van den Brink	**Data and implementation** - Data description **Report** - Introduction - Positive/negative terms in tweets graph - Wordtree - Discussion - Evaluation - Final lay-out corrections
All	- Visualization design decisions - Interaction design decisions - Description of interaction states - Review and revise sections of report

References

Bird, S., Loper, E., & Klein, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc.

Elmqvist, N., & Fekete, J. D. (2010). Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *Visualization and Computer Graphics, IEEE Transactions on*, *16*(3), 439-454.

Graham, L. (2008). Gestalt theory in interactive media design. *Journal of Humanities & Social Sciences*, *2*(s1).

Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., & Ziegler, H. (2008). Visual analytics: Scope and challenges (pp. 76-90). Springer Berlin Heidelberg.

North, C. (2006). Toward measuring visualization insight. Computer Graphics and Applications, IEEE, 26(3), 6-9.

Oghina, A., Breuss, M., Tsagkias, M., & de Rijke, M. (2012). Predicting IMDb movie ratings using social media. In Advances in Information Retrieval (pp. 503-507). Springer Berlin Heidelberg.

Segel, E., & Heer, J. (2010). Narrative visualization: Telling stories with data.Visualization and Computer Graphics, IEEE Transactions on, 16(6), 1139-1148.

Shneiderman, B. (1991). Tree visualization with Tree-maps : A 2-d space-filling approach, 1–10.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages* (pp. 336-343). IEEE.

von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., Van Wijk, J. J., Fekete, J.-D., & Fellner, D. W. (2011). Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum*, 30(6), 1719–1749. doi:10.1111/j.1467-8659.2011.01898.x

Wattenberg, M., & Viégas, F. B. (2008). The word tree, an interactive visual concordance. Visualization and Computer Graphics, IEEE Transactions on, 14(6), 1221-1228.

Yi, J. S., ah Kang, Y., Stasko, J. T., & Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1224-1231.

Appendix A: Data description

This appendix describes the most relevant data retrieved from Twitter and IMDb.

Twitter

The Twitter data can be divided in two separate types of features: surface and textual features. Surface features are features about either the user or the tweet itself, which are subdivisions of surface features themselves. Textual features are features that are pertaining to the actual content of the tweet.

Surface features

User features

user.followers_count
This feature lists the number of followers each account has. This can be used to measure the popularity or influence a user has.
type: ratio

user.friends_count
This feature contains the number of friends each account has. And as with the `followers_count` can be used as a measure of popularity or influence.
type: ratio

user.screen_name
This is the screen name of the account which is used in retweets and tweets directed to other people of entities.
type: nominal

Tweet features

favorite_count
This field contains the number of times a tweet has been favorited by another user. When tweets about a movie are often favorited, it could mean that this movie will be popular.
type: ratio

retweet_count
This feature contains the number of times this tweet has been retweeted by another user. As with `favorite_count`, a tweet about a movie that has a high `retweet_count` could be an indication of popularity or success.
type: ratio

created_at
This is when a tweet is created.
type: interval

Textual features

text

`Text` is the actual content of the tweet and will be treated as a bag of words from which for example a sentiment analysis can be performed.
type: nominal

hashtags

`Hashtags` are all the hashtags taken from the `text` of the tweet and will also be treated as a bag of words. This can be useful for looking up mentions of movies.
type: nominal

user_mentions

`User_mentions` are all the `screen_names` taken from the `text`. This will also be treated as a bag words and can be used for looking up actors, actresses or directors.
type: nominal

IMDb

The data retrieved from IMDb.com consisted of 49 .gz files with a compressed size of 1.14 GB. Inside these .gz files were .list files. These files can be opened using most text editors.

actors.list/actresses.list/directors.list

The first file is called `actors.list` and is immediately the largest of them all (uncompressed it weighs in at 737 MB). This list contains all male entries that were actors in movies and/or were recurring characters in tv shows. In this file all actors that are in the IMDb are listed in alphabetical order with next to their name all the movies and shows they were in, when this movie/show was released/aired, the name of the character they were playing and the billing position in the credits.
Next we have the `actresses.list` file, this file is structured in exactly the same way as actors.list, but then of course only listing actresses and their appearances. Example:

```
Stapel, Huub
           'n stukje humor (2002)  [Gerard]
            Alles is liefde (2007)  (uncredited)  [Himself]
            Amsterdamned (1988)  [Eric Visser]  <1>
```

`Director.list` lists every movie all listed directors directed in the same format.
This information is highly relevant to us because it enables us to compare who was part of previous movies for which we are able to retrieve both the rating and the gross revenue on the opening weekend and the cast of the movie for which we want to predict the gross revenue and rating. This comparison will hopefully give us an indication of the movie's success and rating
type: nominal

aka-names.list/aka-titles.list

This file lists all the actors and actresses with an alternative spelling or word order as their name and lists these below their name. Example:

```
Cabau, Yolanthe
  (aka Kasbergen, Yolanthe Cabau-van)
  (aka Sneijder-Cabau, Yolanthe)
```

The `aka-titles.list` lists all movies and tv serie entries that have an alternative title but it follows the same format with as above. It does, however contain extra information like where and when this title was used. Example:

```
"'t Schaep Met De 5 Pooten" (2006)
  (aka "'t Spaanse Schaep" (2011))    (Netherlands) (third season title)
  (aka "Vrije Schaep Met De 5 Pooten, 't" (2009))    (Netherlands) (second
 season title)
```

These files can help us to acquire Twitter messages where an alternative spelling might have been used instead of just the messages where the name or title that IMDb uses is used.
type: nominal

business.list

This file lists how well a movie or tv show where and when did in counts of admissions (AD), revenue (GR), rentals(RT), opening weekend (OW), what it’s budget (BT) was and the dates when it was filmed (SD). Example:

```
MV: Ace Ventura: When Nature Calls (1995)
AD: 329,421 (Netherlands) (1 January 1997)
BT: USD 30,000,000
GR: USD 209,300,000 (Worldwide)
OW: USD 37,804,076 (USA) (12 November 1995) (2 screens)
RT: USD 49,606,000 (USA)
SD: 27 March 1995 - 28 July 1995
```

This file is also very important as it gives us the first metric we have to try and predict (OW) for older movies. Goal is to try and find out what influences this metric.
type: ratio

certificates.list

Lists the available ratings in different countries for every entry.

```
Die Hard (1988)                           USA:R    (certificate #29160)
```

type: nominal

genres.list

This file lists every entry with one or more of the following genres (also listed the number of appearances in the whole list):

Short	359449	Sci-Fi	20993
Drama	244637	Mystery	20662
Comedy	185366	Biography	17060
Documentary	162119	Sport	15589
Adult	59388	History	14868
Romance	47741	Musical	14394
Action	47676	Western	13517
Thriller	43037	War	12303
Animation	42689	Reality-TV	9901
Family	39127	News	7512
Crime	35046	Talk-Show	6567
Music	31874	Game-Show	4254
Horror	31140	Film-Noir	471
Adventure	29860	Lifestyle	1
Fantasy	24246	Experimental	1

You can imagine certain kind of movies bringing in more money than others. So that might not be very interesting, but you could for example combine this one with the cast or release date to make predictions more accurate.
type: nominal

plot.list

Where available this file lists all submitted plots for movies and tv show episodes.
This can be very useful when made into a bag-of-words to try and see if movies with a certain rating share a list of words. And if they do it is possible to compare this list to the bag of words of the target movie's plot.
type: nominal

ratings.list

Contains the rating, number of votes and vote distribution for (i) the top 250 movies (25000+ votes), (ii) the bottom 10 movies and for (iii) an alphabetical list of all movies.

```
New  Distribution  Votes  Rank  Title
      0000000125  960481   9.2  The Shawshank Redemption (1994)
      0000000125  684710   9.2  The Godfather (1972)
```

Very useful when trying to predict another movie's success as a way to compare it to other movie's ratings.
This list contains the data for the second metric we have to try and predict: the rating. Same as with `business.list` we will try to find out what other variables influence this.
type: ratio

release-dates.list

Contains movies and tv show episodes and where and when they were released.

```
The Shawshank Redemption (1994)            Spain:24 February 1995
The Shawshank Redemption (1994)            France:1 March 1995
The Shawshank Redemption (1994)            Netherlands:2 March 1995
```

It is possible that movies that are released in a certain season get rated higher, or maybe better movies are released in certain seasons. Either way, comparing the release date of an unreleased movie to release dates of previous movies can give us insight in

This information can be used to compare the upcoming movie’s release date to to see if it can be useful when trying to predict the IMDb movie rating and opening weekend revenue. This can of course be combined with for example genre or cast for (hopefully) better predictions.

type: interval

taglines.list

In the file a tagline is described as “a short description or comment on a movie that is displayed on movie posters (or direct to video covers etc) to capture the essence of the movie, and ultimetely(sic) make you watch the movie.”

This file contains the tagline(s) for every movie where available.

```
# Licence to Kill (1989)
   Out for revenge. Glimpse behind the cool facade of 007. And see how sweet
revenge can really be. [USA poster]
   James Bond 007
   His bad side is a dangerous place to be.
   James Bond is out on his own and out for revenge.
```

Same as with the plot, this can be treated as a bag of words for movies with for example a similar cast/actor, release date or genre as the target movie. So a comparison between both bags can be made. Another option would be to put both the taglines and the genres in the same bag if results show that when taken individually they are poor predictors.

type: nominal