Application of Textual Quantification and Analysis for Document Classification

Fadi Almazyad, Russell Davis, Walter Gerych

Term-weighting schemes have applications in a number of fields of information retrieval, including document recommendation systems [1], search engine optimization [2], and user modeling [3]. One of the more prominent schemes is term frequency - inverse document frequency (TF-IDF) weighting.

Term frequency assesses the number of times an n-gram of words appears in a given document. It can be treated as a weighting term for the importance of that n-gram to the document; an n-gram that appears multiple times is treated as more important than one appearing fewer times. Inverse document frequency is a weighting term that diminishes the weight of n-grams that appear frequently in a set of documents. The more often an n-gram appears across the document set, the less importance is ascribed to that term.

Used in tandem, term frequency and inverse document frequency assess the importance of an n-gram to a document that is part of a larger document set -- terms are weighted more for appearance in a given document but weighted less for appearance in the set as a whole.

In order to generate the TF-IDF statistic, we will rely here on the TfidVectorizer class in scikit-learn [4]. This class transforms a text corpus first tokenizing the corpus (creating a vector of n-grams) and counting times those n-grams appear in each document, and then by using the tf-idf transform to change the raw counts into a matrix of weighted features. The TfidVectorizer class can take a variety of parameters as inputs in order to customize the way these operations treat the corpus. To examine instances of TF-IDF transformation, we will use a data set of comprised of 2000 textual movie reviews, of which 1000 were positive and 1000 were negative [5].

The document frequency and ngram_range parameters are some of the more prominently used parameters. The min_df parameter excludes any n-gram that has a document frequency lower than a given threshold, either based on the absolute count of documents it appears in or the proportion of documents it appears in. The default value for min_df is 1, meaning that it includes any term appearing in at least one document, i.e. all terms. Similarly, max_df excludes n-grams with a document frequency exceeding a given threshold, based on absolute document count or proportion of documents. The default value for max_df is 1.0, indicating that any terms that appear in greater than 100% of documents are excluded; in other words, it includes all terms.

Subsequently, increasing the value of the min_df parameter would be expected to reduce the number of terms included in the evaluation, as it exclude terms that are only present in a small number of documents. Similarly, lower values of max_df would also reduce the number of terms included, as it exclude terms that appeared in virtually all documents.

To examine the impact of changes in these parameters, we used the previously established training set. Figure 1 shows the results of various parameters on the feature count in that set. The size of each point corresponds to the natural logarithm of the feature count.

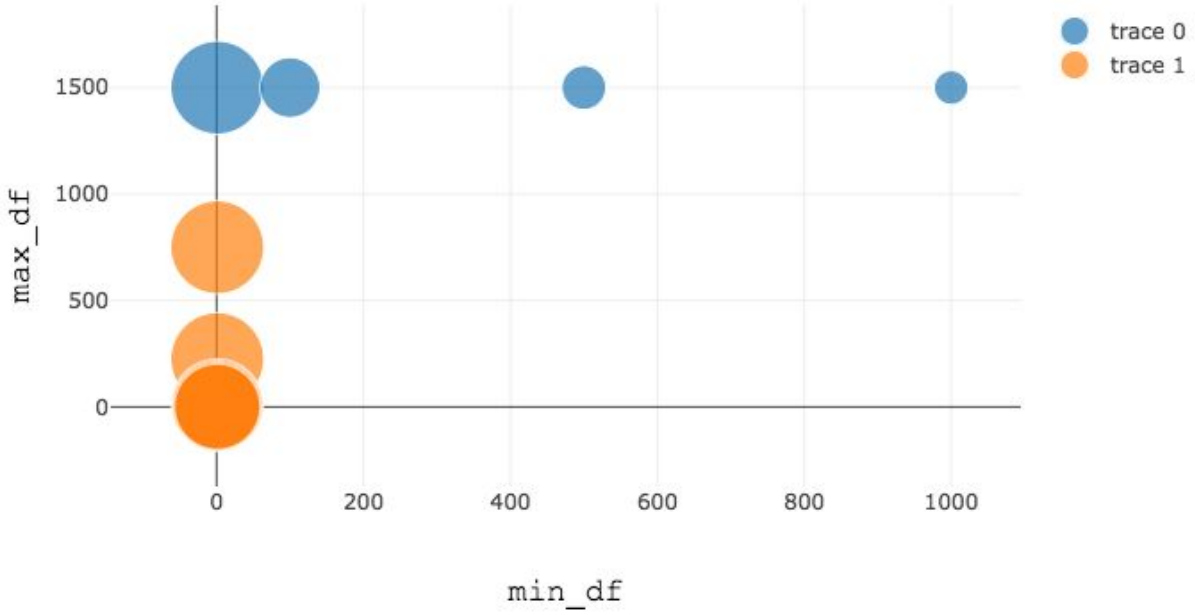

Figure 1

As expected, higher values of min_df lowered the number of included terms, as did lower values of max_df. Most of the terms were excluded when the min_df value was over 100, indicating that a relatively small amount of terms appeared in more than about 10% of the documents. The max_df values did not see meaningful reduction until a max_df value of around 10, and even at a max_df value of 1 the feature count was over 14,000. This suggests that nearly half of the terms appeared only in a single document.

The ngram_range parameter allows customization of the n-grams evaluated by the class. It includes all values within an input range. Large ranges will lead to analysis of n-grams of varying sizes, yielding data that can provide significant insight into frequently occurring phrases. Figure 2 shows the results of varying ranges on the quantity of data generated using default parameters of min_df and max_df, with the feature counts transformed using the natural logarithm.

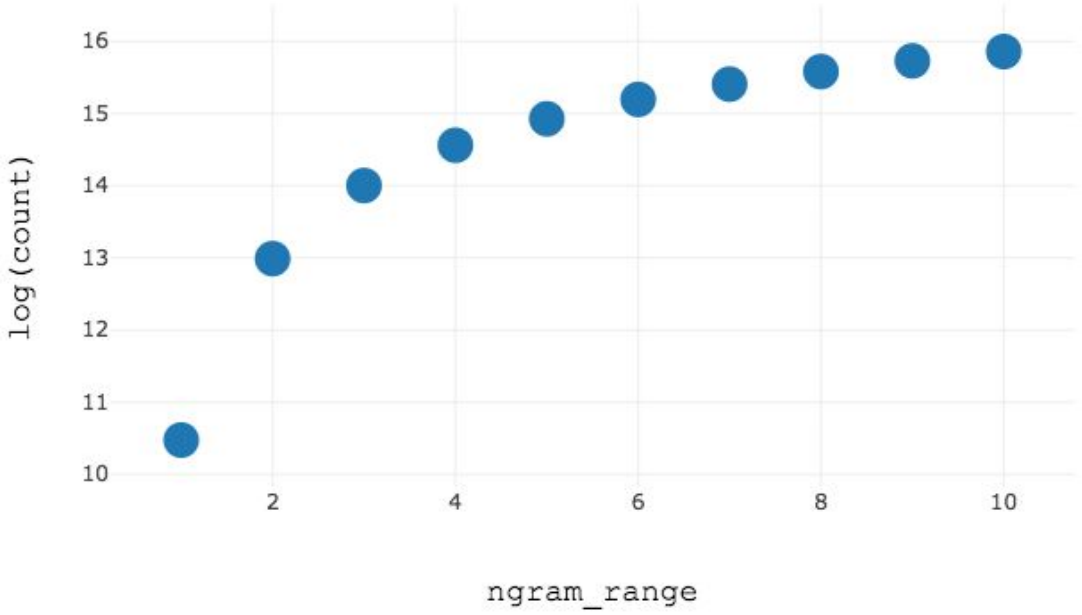

Figure 2

The feature count increases dramatically initially, suggests that n-grams of up to size 3 provide unique information. However, n-grams of size 4 and above increase the cumulative count linearly; this suggests that they do not provide unique information, as they are simply adding words to already-unique phrases. The gain in information is not likely to be worth the increase in complexity by increasing the n-gram range past (1,3).

The matrices generated using the TF-IDF vectorizer contain a wealth of information about the distribution of words among the analyzed corpus of texts. By using a collection of documents to generate an IDF calculation, we developed a TF-IDF weighting scheme. This allowed for analysis of new documents from a similar set based on the existing corpus. After developing the appropriate weighting factors for a range of TF-IDF parameter values, we used them to transform the test data into TF-IDF matrices to test two classification methods.

As a proof of concept, we used Python code prepared by Olivier Grisel as part of the scikit-learn distribution to demonstrate that text data vectorized using TF-IDF could be effectively applied with Linear Support Vector Classification (Linear SCV) [6]. Linear SCV is a supervised binary linear classifier that maximizes the margin between two sets of data; in this case, the two sets are positive and negative textual reviews. After the maximum margin is established, it can be used to predict the class of new data sets, i.e. test data. For this proof of concept, we varied only ngram_range and applied grid search cross-validation to ensure that

the optimal ngram_range was selected. A confusion matrix obtained is presented in the attached code and suggested a classification accuracy of 83%.

To further examine the use of Linear SCV with the training data, we vary the penalty for misclassification, C; this will change the manner in which the maximum margin is established. We investigated the impact of more significant adjustments to TF-IDF parameters and variations in C, we determine the parameters with the highest success rate. Figure 3 shows the TF-IDF vectorizer parameters, penalty for misclassification value, confusion matrix obtained for the most successful set of parameters using Linear SCV.

Minimum DF	Maximum DF	N-gram Range	Classifier Value	Confusion Matrix	Success rate for Testing
1	0.75	1,2	C = 10	195 49 35 221	83.2%

Figure 3

The other classification method tested was K-Nearest Neighbors (KNN) classification. KNN classification is a non-parametric method that predicts a class for test data based on the class of the k nearest data points. By varying the value of K, we adjust the flexibility of the resulting model. Figure 4 shows the TF-IDF vectorizer parameters and confusion matrix obtained for the most successful value of K.

Minimum DF	Maximum DF	N-gram Range	Classifier Value	Confusion Matrix	Success rate for Testing
1	0.85	1,3	n_neighbors = 500, Weights = distance	171 73 40 216	77.4%

Figure 4

Linear SCV is significantly superior to K-Nearest Neighbors classification in accurately predicting whether or not a given test document is a positive or negative review. Because Linear SCV is a linear classifier while KNN is non-parametric, it would be likely to have superior performance when treating any data set that fits this assumption and can be classified across a linear boundary. It seems reasonable to conclude that positive and negative reviews have

distinct differences in word content that make them linearly separable, making Linear SCV's superior performance expected.

To assess the factors that may have contributed to mischaracterization, we will examine Figure 5, which shows the terms with the highest average weight across the corpus of misclassified documents in each category. This figure was inspired by and based on code written by Thomas Buhrmann [7]. The terms listed in Figure X, except for a few exceptions, are not words that can be recognized as positive or negative. TF-IDF analysis does not involve linguistic parsing of individual words or phrases, but focuses only on the raw count of n-grams. Therefore, individual documents that do not share sufficient characteristics with other parts of the corpus can be misclassified if enough terms, including ones that would not be normally parsed as being indicative of positive or negative speech.

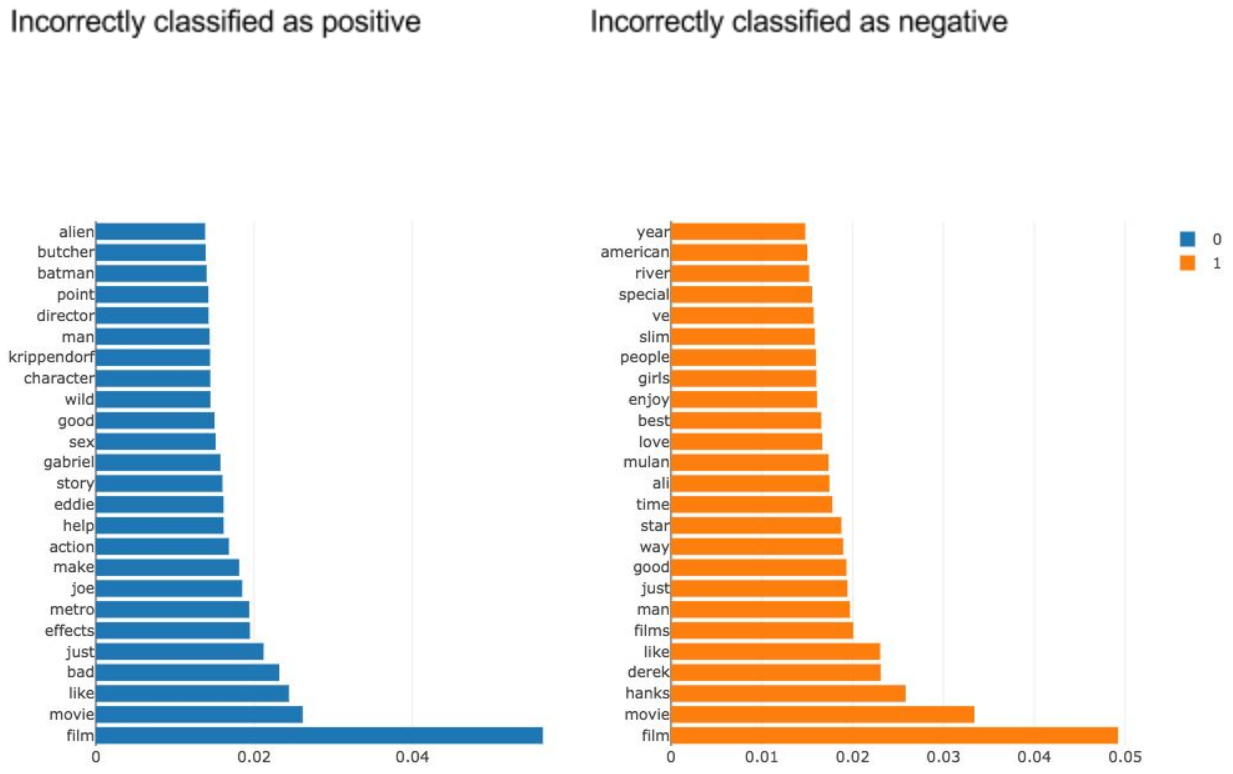

Figure 5

Appendix A contains the text of a pair of randomly selected misclassified reviews, one positive and one negative. In the first, which is a negative review, there is a preponderance of words that can be easily parsed as praise. The phrases that are condemning, such as “the film takes a dive off the high board”, would not be easily parsed by a machine as negative. In the second, there is an absence of words that an English speaker would characterize as positive or negative in order to help assign the review to its appropriate category. While it is possible that

this may not be unique to these reviews, it is illustrative of the difficulties that can occur during categorization. In the absence of words exhibiting clear sentiment, categorization will occur based on the general content of the review, which depending on the focus and phrasing of the reviewer could understandably result in misclassification.

Because the TF-IDF vectorization and subsequent computerized analysis of the reviews takes place in high-dimensional space, it can be difficult from a human perspective to understand the context in which reviews are classified. To address this, we attempted to find ways to display the data in 2-dimensional space to more easily display the relationship of positive and negative reviews.

First, we used a sentiment analysis tool called Textblob to see how easily the reviews could be separated based on semantic interpretation of their content. This tool assigned each document a value between -1 and 1 based on the assessed positivity of that review. Sentiment analysis is specifically designed for use in purposes such as this, so we expect that the separation will be clear. Figure 6 shows the results of sentiment analysis graphed against document length.

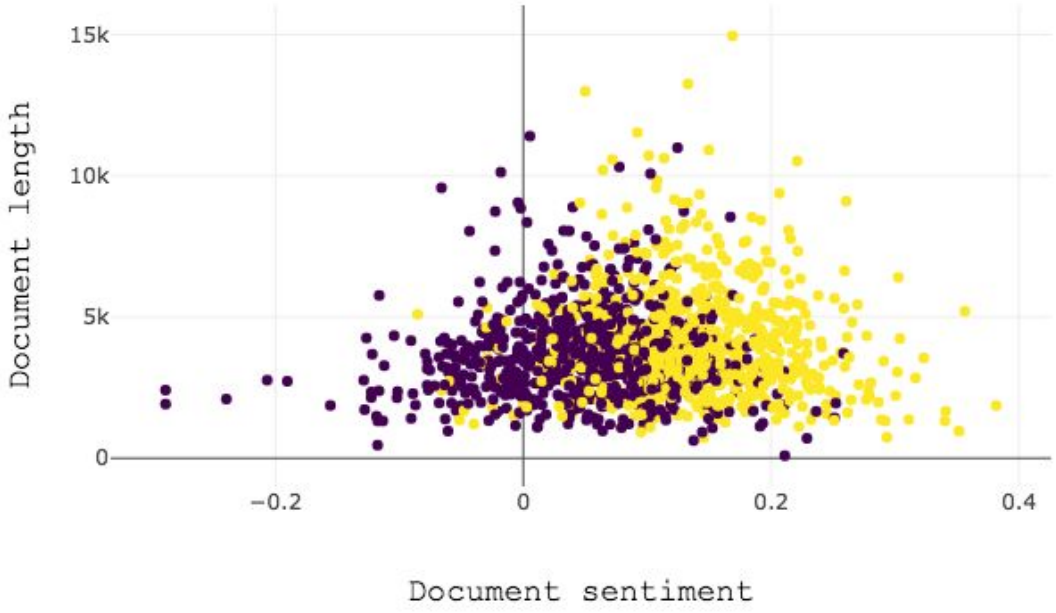

Figure 6

As expected, sentiment analysis does a relatively good job of separating the data. However, it is not easily generalized to other sets of text documents. Use of Textblob and similar tools requires that the corpus being analyzed discuss positive and negative content. If our intent was to separate the reviews by genre rather than positivity, for example, this approach could not

be used. We would like to find an approach that does not require such assumptions about the content of the corpus. Additionally, this approach also does not utilize TF-IDF vectorization, which is necessary for a variety of textual applications, as discussed previously in the report.

A standard approach to simplifying high-dimensional data is Principal Component Analysis, or PCA. PCA finds a small number of dimensions, each of which is a linear combination of variables from the initial high-dimensional space. The dimensions generated by PCA are selected to maximize the sample variance captured. We do not expect that PCA will be too successful due to the assumption of linearity that is integral to PCA. While the corpus as a whole consists of two types of documents that can be distinguished, the vectors are comprised of phrases from the English language, which do not have the linear variation necessary for optimal application of PCA. Figure 7 shows the results of using PCA to generate a two-dimensional plot.

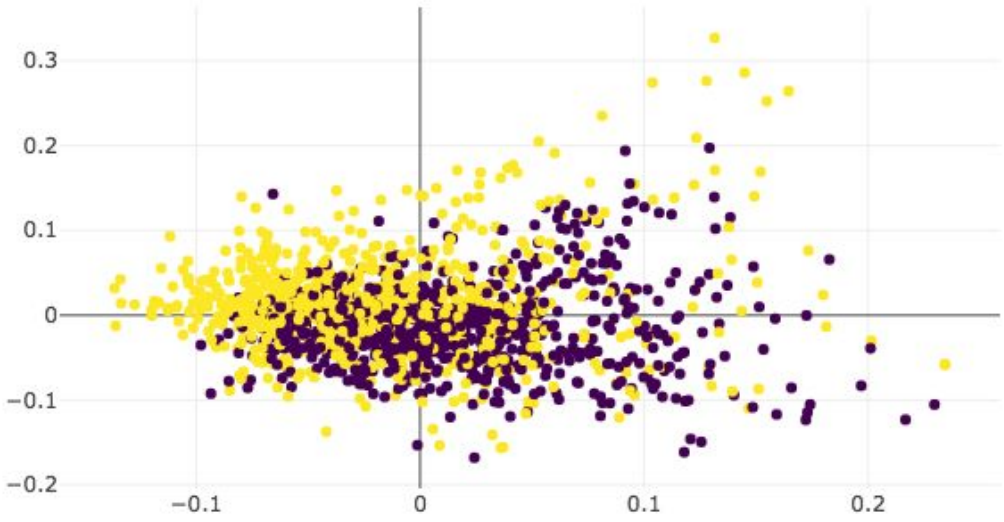

Figure 7

As expected, the separation between positive and negative reviews is clearly less distinct than was achieved for sentiment analysis. Because the primary assumption of linearity made by PCA is not met, the visualized data points are much less distinct than in the plot generated using sentiment analysis..

In order to avoid this pitfall, we turn to nonlinear dimension reduction methods, also called manifold learning. Per their name, these methods do not make the assumption of linearity, and are likely to outperform PCA. The mathematics behind manifold learning are

highly complex, but scikit-learn provides functions that perform the necessary calculations. We briefly experimented with a number of different manifold learning techniques. Figure 8 shows a 2-dimensional plot obtained using a method called spectral embedding, which had better results than many other methods.

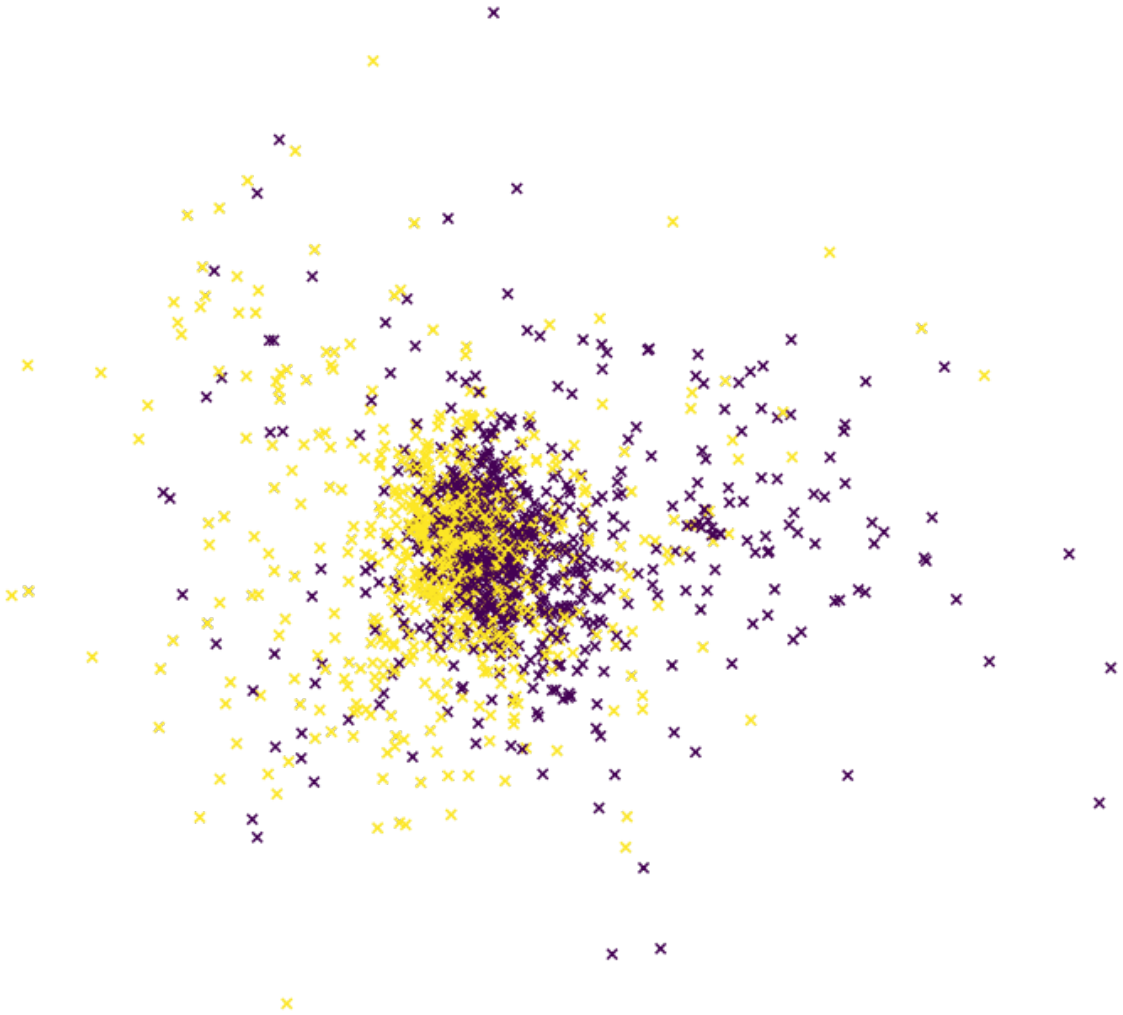

Figure 8

The 2-dimensional plot obtained using spectral embedding is significantly better at separating positive and negative reviews than the one obtained using PCA. Despite the improvement, there is still a large cluster of both positive and negative reviews at the center of the graph, which probably suggests that the word choice for many reviews is similar regardless of whether or not the review is positive or negative.

In the case of textual movie reviews, we have demonstrated that new movie reviews can effectively be classified into positive and negative categories. This could be effectively applied by producers and advertisers in contexts such as social media forums to gauge users' attitudes towards a movie or another product.

These tools can also be used to gauge other sorts of disagreement, such as political dissent. If applied by a heavy-handed government, these tools could be applied to the public discourse in order to quickly identify individuals who do not toe the party line. In the internet age, where massive amounts of communication occurs, the ability to rapidly process text will be a boon to any modern-day Ministry of Love [8].

Our results are indicative of the strength of computers in analyzing high-dimensional data and the difficulties in transforming the results to make them easily understood by humans. In cases where more complete information about the data is available, some of these difficulties can be ameliorated, but it is not always feasible. On the whole, however, it is easy to see the success of TF-IDF vectorization when applied to textual analysis and the variety of potential applications.

Works Cited

[1]https://link.springer.com/article/10.1007/s00799-015-0156-0
[2] https://en.ryte.com/wiki/TF*IDF
[3] https://www.gipp.com/wp-content/papercite-data/pdf/beel17.pdf
[4]http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[5] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[6]https://github.com/scikit-learn/scikit-learn/blob/master/doc/tutorial/text_analytics/solutions/exercise_02_sentiment.py
[7] https://buhrmann.github.io/tfidf-analysis.html
[8] https://en.wikipedia.org/wiki/Ministries_of_Nineteen_Eighty-Four#Ministry_of_Love

Appendix A

Review 1. Misclassified as positive:

“my giant begins with a monologue that's more funny than not and a distinctive `princess bride, ' medieval fairy-tale-in-the- ? 90s feel. i was pleasantly surprised how sharp the comedy was, with very funny scenes occurring on a movie set in romania where talent agent sammy (billy crystal)is visiting a client and in the monastery where sammy ends up after being mysteriously saved when he accidentally plunges his car into a stream. and when sammy meets max, his giant, mysterious savior played by gheorghe muresan, there is magic in the air ; albeit it a bit goofy, i loved every minute of it. the film, of course, plays on the size difference between the two, and at times you would almost swear it must be special effects. i particularly enjoyed a shot of sammy dangling his legs from a max-sized chair as the two face each over an enormous table and eat from oversized bowls and spoons. i also liked jere burns' (from tv's `something so right')performance as the director of the film in romania. at this point, the primary flaw in the film is that gheorghe muresan is incredibly hard to understand, particularly during his fast-paced first scene. it is unfortunate that the script required him to speak quickly right out of the gate, as i did understand him more as the film continued and i became accustomed to his speech.
however, about the time muresan becomes coherent, the characters head to new york and the film takes a dive. off the high board. suddenly, we are expected to believe sammy is a do-whatever-it-takes slime ball. up until this point he was pursuing his own interests to be sure, but he didn't come off as a desperate jerk. if he had, the first scenes could never have been so lighthearted and magical. but now, in new york, we are further introduced to his neglected wife and son, and his character is desperate enough to involve max in a disturbing giant vs. midgets wrestling match, which i found quite unpleasant and jarring within the framework of the film. indeed, many scenes stuck out of this film like incorrectly placed puzzle pieces. this includes the scenes featuring steven seagal, when sammy gets max a role in a big, hollywood movie filming in las vegas. while i enjoyed seeing seagal poke fun at himself, the scenes appeared to exist solely because of his participation, rather than because the film demanded it.
and the audience is required to take a huge leap to believe that max would win the role based on his quotations of shakespeare. at this point i found myself thinking, `we're supposed to believe that someone would cast this guy in a film ? ' then i realized he was, and i was watching it. in the final third, the film undergoes another transformation, this time to unbridled sentimentality. note : major plot points are revealed in this and the final paragraph.
max went to america because sammy promised to reunited him with his childhood love, lillianna, who hasn't seen him since he was a normal-sized kid. her refusal to see max (unbeknownst to him)leads us to an awkward `ends justify the means' scene as sammy's wife serena, played nicely by kathleen quinlan, poses as lillianna. i found this scene offensive. not only did it rely on deception to induce warm, fuzzy feelings from the audience, it reduced max to someone we

should pity and coddle, which i thought was quite undeserved. i think max could have handled the truth, gratuitous illness and all. late in the film we learn max, and in fact all giants, have a heart condition which shortens their lives considerably. this could be an enlightening revelation, but the film seemingly presents it only to justify sammy's transformation into a caring, sensitive guy -- and hey, a great dad and husband, too! `my giant' suffers from a poorly constructed story line and undeveloped characters whose actions are determined by plot points rather than their own internal persuasions. a stronger story with more room for character growth might have been possible if the focus was on max's struggle to be accepted and cast in movies instead of sammy's struggles to get money and become a better person. in this scenario, max's illness could have been a integral part of the film not a story motivator. sentiment and emotion would have followed naturally. instead, we're apparently not supposed to like sammy until the end, but we're not allowed to focus on max. crystal and muresan give adequate and at times enjoyable performances, but in the end, `my giant' left me feeling like i'd been fed gruel from a giant spoon.

Review 2. Misclassified as negative:

urban legend surprised me. based on the last few films the genre has produced (including but not exclusive to the likes of i know what you did last summer, disturbing behavior and the disappointing halloween : h20), i was positive that i was in store for another mildly entertaining but silly and ultimately boring rehash of the scream phenomena. thankfully, legend rose above it's soggy premise to become not only a hip, scary and stylish entertainment, but also what will probably become one of the best films of the year. you're all familiar with the plot ; a bunch of overly horny teenagers get systematically slaughtered by a masked maniac who's identity is not revealed until the closing moments of the film when it turns out to be, you guessed it ; everybody's favorite veteran of the whodunit flicks... the only person you didn't suspect! however, just because legend doesn't really break any new ground as far as literary or even technical achievements go does not mean that it can't qualify as first-class entertainment. it was, believe it or not, one of the funnest times i've had at the movies all year. the thing is, i'm not really sure what made this particular stab (pardon the pun)at the genre seem so fresh and alive. maybe it was the above-par performances by the movie's two leads (jared leto and " cybil " actress alicia witt). or, it could have been the on- target direction by jamie blanks. more likely, though, it was because of the inspired premise. i don't think i need to tell you that a killer hacking people up ala urban folklore is much more frightening than a killer fisherman.
urban legend begins with a sequence that reminded me of the classic scream opener, not so much in plotting as in attention to detail that makes for an absolutely chilling teaser for the rest of the film. a pretty young coed is driving down a deserted road when (gasp!)a gas problem (hee-hee)forces her to stop at an equally deserted gas station for a refill. one problem, though. would you want to stop at a station run by brad dourif ? i thought not. predictably, the attendant ends up coaxing her into the main building to sort out a problem with the credit card company. it's odd, though, when you pick up a phone to realize that nobody's on the other line. naturally, the potential victim escapes into her car and drives off. and the when we least expect it...

whack!!! a decapitation. it turns out that the creepy looking attendant was just trying to warn her about the maniac in the back seat of the car. so who is it this time ? the obvious candidate is the slightly off-kilter professor at the college the girl went to (none other than freddy kruger himself, robert englund), who teaches a class on urban legends. a-ha. or could it be the fame-hungry local journalist (leto) looking for a meaty story to put on his resume ? like all movies of this nature, one of the chief pleasures is trying to guess whodunit. and it turns out to be, of course, the one you didn't suspect. i'm not going to pretend urban legend is anything more than it is, but i must give credit where credit is due. and this flick definitely deserves credit for being what not many other recent horror films have been... fun. * complimentary movie ticket courtesy of valley cinemas in lodi, ca