"The Godfather" vs. *"Chaos"*: Comparing Linguistic Analysis based on On-line Knowledge Sources and Bags-of-N-Grams for Movie Review Valence Estimation

Björn Schuller, Joachim Schenk, Gerhard Rigoll
Technische Universität München, Institute for Human-Machine Communication
D-80333 Munich, Germany, schuller@tum.de

Tobias Knaup
Pingsta, Inc.
1700 Seaport Boulevard, Redwood City, CA 94063, USA, tobi@pingsta.com

Abstract

In the fields of sentiment and emotion recognition, bag of words modeling has lately become popular for the estimation of valence in text. A typical application is the evaluation of reviews of e. g. movies, music, or games. In this respect we suggest the use of back-off N-Grams as basis for a vector space construction in order to combine advantages of word-order modeling and easy integration into potential acoustic feature vectors intended for spoken-document retrieval. For a fine granular estimate we consider data-driven regression next to classification based on Support Vector Machines. Alternatively the on-line knowledge sources ConceptNet, General Inquirer, and WordNet not only serve to reduce out-of-vocabulary events, but also as basis for a purely linguistic analysis. As special benefit, this approach does not demand labeled training data. A large set of 100 k movie reviews of 20 years stemming from Metacritic is utilized throughout extensive parameter discussion and comparative evaluation effectively demonstrating efficiency of the proposed methods.

1 Introduction

Emerging new Internet technologies such as blogs or review websites encourage users to post their own views on products, news articles, or movies. While a lot of effort has been put into estimating valence of product reviews, movies have had less attention in the past. This might be due to the fact that movie reviews are more difficult to handle than e. g. product reviews. Turney [9] observed a discrepancy between the orientation of words that describe the elements and the style of a movie, leading to only 66% accuracy for movies in contrast to up to 84% for automobile reviews. Pointwise mutual information is used to determine valence. The data set consists of 410 reviews from different domains. Pang et al. [5] compare different machine learning techniques and word level features for sentiment classification of movie reviews on a corpus of 1 400 reviews. Best results are achieved with Support Vector Machines (SVM) using word presence information as features. Word frequency, N-grams, part-of-speech (POS), and word position information do not improve performance in their case. A method based on multiple knowledge sources and grammatical patterns is described in [12]. Features and opinion words are learned from training data, and the latter are enhanced by facilitating WordNet. Feature-opinion pairs are then built using grammatical patterns. Experiments are carried out on a corpus of 1 100 reviews. In [1], context-dependent opinion words are utilized in addition to general ones. A number of linguistic rules are used to associate detected opinions to topic features. Liu et al. [4] introduced a novel affect sensing system based solely on world knowledge about everyday situations. The contributions of this paper lie in two fields: First, to the knowledge of the authors, the largest annotated corpus of movie reviews so far is presented, containing over 100 k instances. Experiments with both machine-learning and linguistic methods are carried out for the first time on a movie review database of that size. Second, on-line knowledge sources are incorporated into both methods for improved accuracy and attempt to resolve known issues. Additionally, we show how a regression approach can resolve more subtle differences than *"The Godfather"* – the best rated movie of the database – vs. *"Chaos"* – on the lowest end.

2 Metacritic Database

The database used is the Metacritic film and video review corpus. Metacritic[1] is a website that compiles reviews from

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

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.194

various sources for films, video/DVDs, books, music, television series, and computer games. An automated crawler was used in order to retrieve the HTML source of the web pages and store it for further processing. In order to narrow down the amount of data we use only film and video/DVD reviews in our experiments. A total of 102 622 reviews for 4 901 movies were downloaded.

Reviews are stored as excerpts of key sentences taken from the original text. The average length of a review is 1.4 sentences, with a standard deviation of 0.7. Thus, Metacritic contains mostly short statements rather than long texts. The average length in words is 24.2, ranging between 1 and 104, with a standard deviation of 12.4. Each textual review is accompanied by an integer score value ranging from 0-100, with higher scores indicating a better review. Metacritic scores are calculated from the original numeric rating scheme used by each source. This property, and the huge number of reviews make Metacritic an ideal corpus for sentiment analysis: the accuracy of machine learning algorithms depends on the amount of available training data. Usually, a significant amount of manual annotation work is required to build a corpus that is large enough to yield satisfying results. However, since the Metacritic score can be directly used for regression or classification, no further annotation is needed.

Metacritic has its own classification schema that maps scores into five categories of meaning, and three different colors. Depending on the subject type, different mappings are used. The mapping for movies, books, and music is shown in the following table[2]:

Meaning	Score	Color	Reviews
Universal Acclaim	81 - 100	green	15 353
Generally Favorable	61 - 80	green	38 766
Mixed or Average	40 - 60	yellow	32 586
Generally Unfavorable	20 - 39	red	13 194
Overwhelming Dislike	0 - 19	red	2 723

Table 1. Mapping of score to meaning and color for movies, books, and music.

Since our experiments focus on determining discrete valence values first, we rely on the color coding to distinguish between positive, neutral, and negative reviews. As opposed to this regression is performed on the full range of score values. The number of reviews in each class differs largely, as can be seen in table 1: there is a more than three times larger number of positive than negative reviews.

The vocabulary of the database has a size of 83 328 different words. Looking at POS information, nouns are the most frequent ones (46 563), followed by verbs (12 623), adjectives (9 356), and adverbs (4 032). We used the *ParserTagger* from the OpenNLP software package, a POS tagger that uses maximum entropy prediction.

3 Linguistic Analysis

3.1 On-line Knowledge Sources

ConceptNet is a semantic network, which contains common sense knowledge in a machine-readable format. Concepts are interlinked by 26 different relations that encode the meaning of the connection between them, e. g. `IsA` or `UsedFor`. An assertion consists of two concepts and one relation, e. g. *a cinema* `UsedFor` *to watch a movie* (*"You can use a cinema to watch a movie"*). To account for the fact that in the majority of cases assertions are not true if the order is changed (*"You can watch a movie to use a cinema"* does not make sense), relations are always unidirectional.

General Inquirer is a lexical database that uses tags to carry out its tasks. Each entry consists of the term and a number of tags denoting the presence of a specific property in the term. The two tags `Positiv` and `Negativ` express valence, hence they are the most interesting for our task. There is only partial support for POS information, to the extent that some terms have tags for different verb or adjective classes assigned.

WordNet is a lexical database that organizes lexical concepts in sets of synonymous words, called *synsets*. Unlike ConceptNet, synsets are not linked by relations that express common sense knowledge. They are rather connected by their lexical or semantic relatedness, such as meronymy (one word being a constituent part of another one), or antonymy (one word being the opposite of another). However some of these relations are also found in ConceptNet, e. g. the complement of meronymy is `PartOf`.

3.2 Algorithm Description

As a preprocessing step, the text is split into sentences, which are in turn analyzed by a syntactic chunker to label words and phrases with corresponding part-of-speech (POS) information. As a unit of sentiment representation, we extract *ternary expressions (T-expressions)* on a per-sentence basis. T-expressions were introduced by Katz [3] for automatic question answering tasks, and have been adapted to product review classification by Yi et al. [11]. In the latter context, a T-expression has the following format: `<target, verb, source>`. Target refers to a feature term of the topic the text is about, i. e. a movie in our case. The verb and source, which is typically an adverb, are extracted from the context of the target. Thus, the T-expression for the phrase *"a well written story"* is

[2] http://www.metacritic.com/about/scoring.shtml

`<story, written, well>`. However there are common situations where verbs are absent from a phrase, e. g. *"a great movie"*. Since a T-expression cannot be built in this case, we fall back to a second form, referred to as *binary expression (B-expression)* as suggested in [11]. B-expressions are simply a combination of an adjective and a target co-occurring in the text. Hence, the aforementioned utterance is represented by `<movie, great>`. Target candidates are extracted from noun phrases (NP), based on the observation that feature terms are nouns. To further limit the search space, we use the base noun phrase (BNP) patterns described in [11]. All mentions of the personal pronoun "it" are regarded as referring directly to the topic of the text, and are also used as targets.

In the next step, the goal is to identify sentiment sources for each target, i. e. words that convey the actual affect information. In order to asure that a given sentiment source is actually directed to the target in question, we need to restrict the search space. We accomplish this by finding "border indicators" that appear between clauses or phrases within a sentence, and thus, split it into units of statements. A sentiment source is only associated with a given target if they both occur in the same section, i. e. there is no border indicator between them. Currently, subordinating conjunctions, prepositional phrases, coordinating conjunctions, and punctuation such as commas or colons are used as border indicators. We select all verbs and adjectives from the target section, and use General Inquirer to determine the sentiment valence. In case a word is not found in General Inquirer, we use WordNet synsets to also lookup a word's synonyms, until a match is found. We refer to words for which a valence was found as "sentiment words". Depending on the POS of the sentiment word, T-expressions and B-expressions are built. It appears intuitive that words that occur closer together are more related. Thus, for each sentiment source, only the expression with the shortest distance between its sentiment source and its target is kept for further processing. However multiple expressions may exist for a given target to catch all sentiment words in phrases such as *"a well written, beautifully directed story"*. Furthermore, we also use the word distance to boost or lower the valence of an expression by means of a decay function. The weighted score s of an expression that contains the target t_i and the sentiment source u_i is calculated according to the following equation:

$$s(u_i, t_i) = c \cdot v(u_i) \cdot \frac{1}{D(u_i, t_i)^e} \quad (1)$$

The valence of u_i, as taken from General Inquirer, is denoted by $v(u_i)$, and $D(u_i, t_i)$ is the distance of words between u_i and t_i. Because $\frac{1}{D(u_i,t_i)^e} = 1$ holds for any e if $D(u_i, t_i) = 1$, i. e. u_i and t_i occur right next to each other, $c > 1$ effectively amplifies the valence in that case. On the other hand, c has little effect for $D(u_i, t_i) \gg 1$, which is the case for words that occur further apart.We found that the best results are gained by setting $c = 1.0$ and also $e = 0.3$.

Due to the fact that it is not always possible to find an expression in a sentence, we have two fallback mechanisms. The first one is used in case no target could be found. We simply assume that the sentence is referring to the movie in question, and build pseudo B-expressions having "it" as target. Since no word distance information is available, the decay function cannot be applied, resulting in a valence of either +1 or -1 for the sentence. The second mechanism takes place if no target, and no sentiment words could be found. Based on the observation that there are more than three times as many positive than negative reviews, the valence of a sentence is set to +1 as a last resort. However, this case is rare as only 1.9% of the reviews in our test set do not yield any expressions.

The final step of our algorithm determines if the expressions that were found earlier are actually referring to the movie. To that end, we rely on ConceptNet to identify features of a movie, and then filter out expressions whose target is not a feature. In case no expressions remain for a review, the step is reverted. Feature terms are selected by looking up the following assertions in ConceptNet, *feature* being the feature term: *feature* `PartOf` *"movie"*, *feature* `AtLocation` *"movie"*, *"movie"* `HasProperty` *feature*.

4 Data Driven Approach

As primary technique for classification, Support Vector Machines (SVM) are used. We use bag of words [2] to represent text in a numeric feature space. Each feature thereby represents the occurrence of a specific word in a sentence. In previous works, we successfully ported this principle to the field of emotion [8] and interest [7] recognition from text and speech where it more recently became a popular approach.

In order to reduce the amount of features in a meaningful way, stemming is applied, and a minimum occurrence frequency f_{min} is set to discard very rare words. The occurrence frequency, referred to as term frequency (TF), can be transformed in various ways ([10], p. 311, [2]). A common method is the appliance of the logarithm to compensate linearities. Another measure that is widely used for document retrieval is the inverse document frequency transformation (IDF). The idea is that a sentence is basically characterised by words that often appear in it, except that words used in almost every sentences are useless as discriminators. The TF and the IDF transform can be combined, resulting in the TFIDF transform. Additionally the final feature vector for each review can be normalized to the same Euclidean length.

A simple extension of the bag of words idea allows for exploiting linguistic context information as well and is referred to as bag of N-grams. The main difference is the observation of a series of consecutive words as semantic units of interest. The approach allows to observe several N-grams together, determined by a minimum N-gram length g_{min} and a maximum N-gram length g_{max}, similar to "backing-off". All resulting N-grams are equally used as features.

5 Experiments

All experiments are carried out on the same training and test subsets of the Metacritic database. Since the release year is recorded for every movie, we chose to use odd years for training, and even years for the test set. As mentioned in section 4, we set a minimum term frequency $f_{min} = 2$ to remove unimportant features. In order to make the large feature space computable, features only occurring once are discarded after every 25% of the available data has been processed. We utilize an SVM implementation based on the sequential minimal optimization algorithm described in [6]. A polynomial kernel function with a complexity parameter of $C = 1$, and degree $d = 1$ was used.

5.1 Parameter Tuning

Table 2 gives the accuracy obtained by different combinations of transformations and normalization. A simple N-gram frequency (f_{ij}), logarithmic term frequency transformation (TF), inverse document frequency transformation (IDF), and normalization (norm) are considered. The N-gram parameters are set to $g_{min} = 1$ and $g_{max} = 3$ for these experiments.

Transformation	Accuracy	weighted F-measure
f_{ij}	76.53%	78.07%
norm(f_{ij})	76.63%	78.13%
TF	76.90%	78.40%
norm(TF)	76.85%	78.59%
IDF	76.53%	78.07%
norm(IDF)	77.16%	78.59%
TFIDF	76.89%	78.40%
norm(TFIDF)	**77.33%**	**78.74%**

Table 2. Influence of transformations and normalization on bag of N-grams.

Although the values do not differ much, normalization combined with TFIDF yields best results. IDF seems to benefit largely from normalization as it only adds to performance if values are normalized, too.

To investigate the influence of the N-gram length, different combinations of g_{min} and g_{max} are tested. As shown in table 3, the best accuracy is obtained for $g_{min} = 1$ and $g_{max} = 3$, and significantly decreases for $g_{min} > 1$. This leads to the conclusion that single words convey important valence information in the context of movie reviews in Metacritic.

g_{min}	g_{max}	Accuracy	weighted F-measure
1	1	75.61%	77.17%
1	2	76.76%	78.22%
1	**3**	**77.33%**	**78.74%**
1	4	76.46%	77.89%
1	5	76.91%	78.30%
2	2	69.43%	71.67%
2	3	70.65%	72.74%
2	4	71.16%	73.19%
2	5	72.45%	74.24%
3	3	70.92%	72.73%
3	4	71.23%	72.98%
3	5	71.32%	73.06%

Table 3. Accuracy achieved by different minimum and maximum N-gram lengths.

5.2 Out of Vocabulary Events

If a given word contained in the test set does not occur in the training vocabulary, no numerical value is available for determining its valence, and possibly meaningful words are discarded. Compared to the training and test vocabulary sizes (35 259 and 36 645 words, respectively), the number of these out of vocabulary words is quite significant, adding up to 15 525. To overcome this deficiency, we leverage the on-line knowledge sources ConceptNet and WordNet which were introduced in section 3.1 to find synonyms for missing words. If one of the synonyms does occur in the training vocabulary, the original word is replaced with it. This step is carried out before N-grams are assembled. While we were able to substitute 3 897 vocabulary words using this method, the resulting gain in accuracy is only 0.02%.

5.3 Data vs. Knowledge Source driven

To compare the performance obtained by bag of N-grams to the approach based on on-line knowledge sources, we evaluated the algorithm described in section 3.2 on the same test set. The parameters for bag of N-grams are chosen as $g_{min} = 1$ and $g_{max} = 3$ to produce best results. Normalization and TFIDF as well as out of vocabulary resolution are used. The Results are shown in table 4.

Bag of N-Grams clearly outperform the on-line knowledge based approach. Interestingly enough though, the precision measure for negative valence is far below the positive one for both methods, meaning that both methods tend to detect negative sentiment when it is actually positive. This confirms the hypothesis brought up by [9] that positive movie

Measure [%]	BoNG	OKS
Accuracy	77.33%	68.61%
Weighted F-measure	78.74%	69.77%
Precision positive	92.18%	82.08%
Precision negative	50.78%	36.06%
Recall positive	77.00%	75.61%
Recall negative	78.41%	45.46%

Table 4. Bag of N-Grams (BoNG) and On-line Knowledge Sources (OKS) compared

reviews often contain negative wording, e. g. to describe unpleasant scenes in a war or horror movie.

5.4 Regression

In order to obtain a higher resolution, Support Vector Regression (SVR) is used to predict the score value. The SVR margin is set to $\epsilon = 0.001$, and the complexity parameter to $C = 1$. We use a radial basis function kernel with the variance parameter set to $\gamma = 0.01$. Normalization and TFIDF as well as out of vocabulary resolution are used. N-gram parameters are chosen as $g_{min} = 1$ and $g_{max} = 3$. We achieve a correlation coefficient of 0.5626, and mean absolute error of 14.40. This error is partly owed to the fact that the score is usually assigned by the author of a review, and therefore suffers from a certain degree of subjectivity. Also, scores are unevenly distributed within the data set, causing a lack of training instances for some scores.

6 Conclusion

Based on the suggested methods we could resolve more subtle differences than *"The Godfather"* vs. *"Chaos"*. Despite the fact that both of the presented approaches do not rely on features specific to the movie review domain, the achieved accuracy is considerably high. However, our knowledge sources based method is suffering from poor performance for negative reviews. Best accuracy is achieved by using bag of N-grams, in contrast to the results in [5], where single word features yield best performance. We employed on-line knowledge sources as main methods in a linguistic approach, as well as for resolving out of vocabulary events in a data-driven approach. A significant amount of out of vocabulary events could be resolved using this method.

In future work, we will investigate the use of character N-grams compared to word level N-grams. Character N-grams have recently proven to be suitable for general spoken document retrieval tasks and thus appear promising for movie review valence estimation as well. We are also looking to improve overall performance by fusing the linguistic and the data-driven approach. Finally, we will extend the methods in order to estimate mixed/neutral sentiment as well.

7. Acknowledgement

The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211486 (SEMAINE).

References

[1] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *Proc. of WSDM '08*, pages 231–240, New York, NY, USA, 2008. ACM.

[2] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. of ECML-98*, number 1398, pages 137–142. Springer, 1998.

[3] B. Katz. From sentence processing to information access on the world wide web. In *AAAI Spring Symposium on Natural Language Processing for the WWW*, pages 77–86, 1997.

[4] H. Liu, H. Lieberman, and T. Selker. A model of textual affect sensing using real-world knowledge. In *Proc. of IUI '03*, pages 125–132, New York, NY, USA, 2003. ACM.

[5] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proc. of EMNLP '02*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[6] J. C. Platt. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[7] B. Schuller, N. Köhler, R. Müller, and G. Rigoll. Recognition of interest in human conversational speech. In *Proc. INTERSPEECH 2006, ICSLP, Pittsburgh,USA*, pages 793–796. ISCA, 2006. 17.-21.09.2006.

[8] B. Schuller, R. Müller, M. Lang, and G. Rigoll. Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles. In *Proc. of Interspeech 2005 - Eurospeech, Lisbon, Portugal*, pages 805–808. ISCA, 2005.

[9] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 417–424, Philadelphia, July 2002.

[10] I. H. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, second edition, 2005.

[11] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In *Proc. of ICDM'03*, pages 427–434, November 2003.

[12] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *Proc. of CIKM '06*, pages 43–50, New York, NY, USA, 2006. ACM.