Enhanced Sentiment Classification of Telugu Text using ML Techniques

by

Muku Sandeep, Nurendra Choudhary, Radhika Mamidi

in

The 25th International Joint Conference on Artificial Intelligence IJCAI-16

Hilton, New York City, USA

Report No: IIIT/TR/2016/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2016

Enhanced Sentiment Classification of Telugu Text using ML Techniques

Sandeep Sricharan Mukku
LTRC, IIIT Hyderabad
sandeep.mukku@research.iiit.ac.in

Nurendra Choudhary
LTRC, IIIT Hyderabad
nurendra.choudhary@research.iiit.ac.in

Radhika Mamidi
LTRC, IIIT Hyderabad
radhika.mamidi@iiit.ac.in

Abstract

With the growing amount of information and availability of opinion-rich resources, it is sometimes difficult for a common man to analyse what others think of. To analyse this information and to see what people in general think or feel of a product or a service is the problem of Sentiment Analysis. Sentiment analysis or Sentiment polarity labelling is an emerging field, so this needs to be accurate. In this paper, we explore various Machine Learning techniques for the classification of Telugu sentences into positive or negative polarities.

1 Introduction

Recently there is a proliferation of World Wide Web sites that emphasizes user-generated content as users are the potential content contributors. "What people think and feel" - is the important information for marketing and business operations as it makes their product or service better. Also, there are a lot of comments and blog-posts about trending activity in social media. People try to analyse this information and try to draw conclusions out of them. To better analyse and classify this information, researchers these days are actively working on sentiment analysis. Sentiment Analysis or polarity classification is an effort to classify a given text into polarities, either positive or negative. Majority of the work in the field of sentiment classification has been done in English. There has been very less contribution for regional languages, especially Indian Languages.

Telugu is a Dravidian language native to India. There are about 75 million native Telugu speakers. Telugu ranks fifteenth in the Ethnologue list of most-spoken languages worldwide[1]. Currently there are a lot of websites, blogs etc., rich in Telugu content. In our work, we tried to classify the polarity of Telugu sentences using various Machine Learning Techniques viz., Naive Bayes, Logistic Regression, SVM (Support Vector Machines), MLP (Multi Layer Perceptron) Neural Network, Decision Trees and Random Forest. We built models for two classification tasks: a binary task of classification of sentiment into positive and negative polarities and a ternary task of classification of sentiment into positive, negative and neutral polarities. The algorithm and formulation are explained in detail in later sections.

The rest of the paper is organised as follows. In section 2, we discuss the previous works and related work. In section 3, we describe the datasets used for our work. In section 4, we discuss about the methodology used in our paper which includes pre-processing, training and output. In section 5, we present the framework of our work which includes the tools and different Machine Learning techniques used in our work. In section 6, we present our experiments and discuss the results. Later, we conclude and discuss the future directions of this work.

2 Related Work

Sentiment classification is a difficult task and a lot of research has been done in the past. In this section we survey some of the methodologies and approaches used to address the task of sentiment analysis and polarity classification. Our work is motivated by most of these works.

Enhanced Naive Bayes model is used for sentiment classification task in English [Narayanan *et al.*, 2013]. Their approach is a combination of methodologies like effective negation handling, feature-selection by mutual information and word n-grams. This resulted in significant improvement of accuracy.

Learning word vectors for sentiment analysis is a research work, where Logistic Regression classifier is used as a predictor. [Maas *et al.*, 2011] proposed a methodology which can grasp both continuous and multi-class sentiment information as well as non-sentiment annotations.

[Mullen and Collier, 2004] uses support vector machines (SVMs) to bring together diverse sources of potentially pertinent information, including several favorability measures for phrases and adjectives and, where available, knowledge of the topic of the text. Predicting the helpfulness of online reviews is another area where [Lee and Choeh, 2014] uses a back-propagation multilayer perceptron neural network. This work motivated us to use multilayer perceptron (MLP) neural network for the task of sentiment classification.

Distributed Representations of Sentences and Documents is the work by [Le and Mikolov, 2014] where they make fixed length paragraph vectors or sentence vectors which are

[1] http://www.ethnologue.com/statistics/size

quite useful for our work. We used the tool Doc2Vec for pre-processing the data. Further usage is explained in detail in later sections of the paper.

[Das and Bandyopadhyay, 2010] propose several computational techniques to generate sentiment lexicons in Indian languages (which includes Bengali, Hindi and Telugu languages) automatically and semi-automatically. [Das and Bandyopadhyay, 2011] proposes a tool Dr Sentiment where it automatically creates the PsychoSentiWordNet involving internet population. The PsychoSentiWordNet is an extension of SentiWordNet that presently holds human psychological knowledge on a few aspects along with sentiment knowledge.

3 Dataset

In this section, we describe the raw corpus and annotated data which are domain independent. These have been used in our experiments.

3.1 Raw Corpus

A corpus consisting of 7,21,785 raw Telugu sentences was provided by Indian Languages Corpora Initiative (ILCI)[2]. These sentences were used for training the Doc2vec model (as described in the next section) for generating sentence vectors.

3.2 Annotated Data

The corpus consists of Telugu sentences each attached with a corresponding polarity tag. There are about 1644 sentences which consists of 1068 positive, 219 negative and 357 neutral sentences. These sentences are used to train, test and evaluate the classifier models.

The corpus is prepared from raw data taken from the Telugu Newspapers[3]. This newspaper raw data was first annotated by two native Telugu speakers separately. The data was then merged by a third native speaker who also validated it simultaneously. The annotation consists of three polarity tags i.e; Positive, Negative and Neutral.

We performed inter-annotator agreement using Cohenś kappa coefficient[4]. We got the annotation consistency (k value) to be 0.92 (which is in perfect agreement).

4 Methodology

In this section we explain the steps involved in our approach. Doc2Vec tool (*Refer section 5.1*)gives the semantic representation of a sentence with respect to a dataset. This means that the vector of the sentence represents the meaning of the sentence. Therefore, classifying the semantic space according to training data can classify all the future instances of the same kind thus giving the solution to the problem of sentiment analysis.

4.1 Pre-processing

We converted the annotated data of sentences to 200-dimension feature sentence vectors. For this we used the Doc2vec tool provided by Gensim[5], a python module.

[2]http://sanskrit.jnu.ac.in/ilci/index.jsp

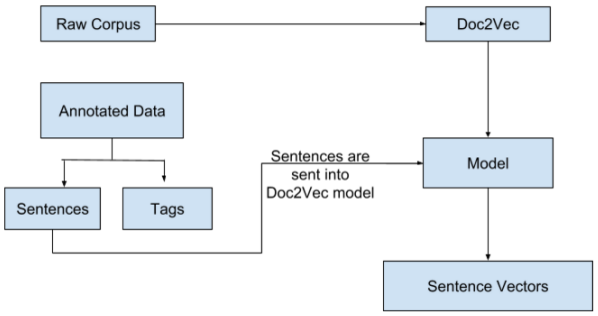
[3]http://www.w3newspapers.com/india/telugu/

[4]http://en.wikipedia.org/wiki/Cohen%27s_kappa

[5]https://radimrehurek.com/gensim/index.html

Doc2vec takes a raw corpus as input and gives us a distributional semantic representation of sentences accordingly. A Doc2vec model is trained on the raw corpus (*Refer section 3.1*). The sentences alone are taken from annotated data and passed through the trained Doc2Vec model. The model then returns sentence vectors for each of the sentences. Here we maintained the correspondence while converting between sentences and their tags.

Figure 1: Data Pre-processing

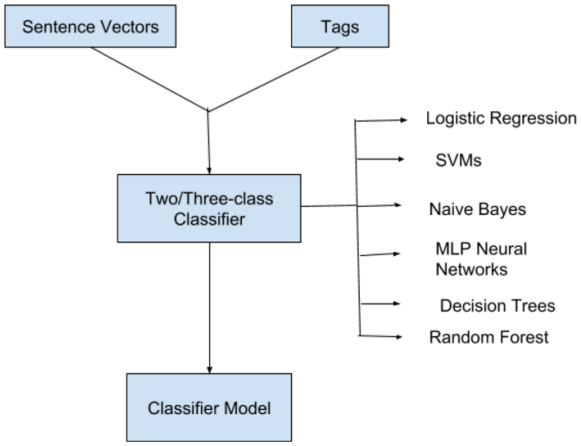


4.2 Training

In the pre-processing phase we converted each sentence of the annotated data into a sentence vector. Therefore we have a sentence vector with a corresponding tag attached to it. Hence the task is reduced to a binary or ternary classification problem. For this task we use various Machine Learning classifiers. The algorithms are explained in the following section.

The model for the classifiers are trained using sentence vectors and their corresponding tags. The models are evaluated using 5-fold cross validation where we divided the data into training and testing sets in the ratio 4:1. The model thus obtained is now ready to classify any sentence vector.

Figure 2: Training

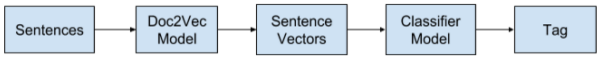


4.3 Output

In this section we discuss the final pipeline which gives the resultant tag for a given input Telugu sentence. The given input

sentence is converted into a sentence vector using a Doc2Vec model. This sentence vector is given to the trained classifier model which returns the output tag.

Figure 3: Output



5 Framework

In this section, we explain the tool used and the various Machine Learning Techniques employed.

5.1 Doc2Vec Tool

Sentence Vector is an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences. In the paper [Le and Mikolov, 2014], their algorithm represents each document by a dense vector which is trained to predict words in the document. Machine learning algorithms typically require the text to be represented as a fixed vector. Usually the most common fixed-length vector representation for texts is bag-of-words (BOW) or bag-of-n-grams [Harris, 1954]. These representations are used because they are simple and accurate. We are not using bag-of-words because this technique has many disadvantages. The word order is lost, and thus different sentences with the same set of words will have exactly the same representation. Also, we did not use bag-of-n-grams because bag-of-n-grams considers the word order in shorter context but it suffers from the curse of higher dimensionality and data sparsity. We found many advantages of sentence vectors such as learning from unlabeled data. Sentence vectors also take into consideration the word order. Doc2Vec is a tool in which sentences are converted into sentence vectors. This tool helps in pre-processing and training of data.

5.2 ML Techniques

We used scikit-learn[6] toolkit which has all these techniques pre-implemented.

Naive Bayes

Naive Bayes (NB) classifier is a probabilistic classifier which uses Bayes Theorem. This classifier evaluates the probability of an event given the probability of another event which has previously occurred. Naives Bayes classifier works very effectively for linearly separable problems. It also works fine for non-linearly separable problems.

Logistic Regression

Logistic Regression (LR) is a multi-class logistic model which is used to estimate the probability of a response based predictor variables in which there are one or more independent variables that determine an outcome. The expected values of the response based predictor variable are formed based on combination of values taken by the predictors. We took the C value (i.e. the regularization parameter) as 1.0.

[6]http://scikit-learn.org/stable/

Support Vector Machine (SVM)

SVM classifier is a supervised learning model which constructs a set of hyperplanes in a high-dimensional space which separates the data into classes. SVM is a non probabilistic linear classifier. SVM models are closely related to a Neural Network. SVM takes the input data and for each input data row it predicts the class to which this input row belongs.

Multi-Layer Perceptron (MLP) Neural Network

A multilayer perceptron (MLP) is a feed-forward artificial neural network model which maps input data sets on an appropriate set of outputs. MLP consists of multiple layers of nodes in a directed graph , each layer is fully connected to the next layer. Feed-forward means the data flows only in one direction, in our case from input to output i.e., in forward direction.

Decision Trees

Decision tree (DT) is a decision support tool that uses a tree-like model for the decisions and likely outcomes. A decision tree is a tree in which each internal (non-leaf) node is labeled with an input feature. Each leaf of the tree is labeled with a class. But for our work decision trees give less accurate results because of overfitting of training data. We took the tree depth as 20 for each decision tree.

Random Forest

Random Forest (RF) is an ensemble of Decision Trees. Random Forests construct multiple decision trees and take each of their scores into consideration for giving the final output. Decision Trees tend to overfit on a given data and hence they will give good results for training data but bad on testing data. Random Forests reduces overfitting as multiple decision trees are involved. We took the n_estimator parameter as 100.

Adaboost Ensemble

The core principle of Adaboost (A B) is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction.

6 Experiments and Results

The method of 5-fold cross-validation is used. The experiments are performed four times (trails) to improve the validity of the results. In each experiment, the sentences in data are chosen randomly for the division into parts. These experiments are performed in the Training Step (*See Fig.2*).

The results are given below as tables. As can be observed for binary classification Random Forest, Logistic Regression and Support Vector Machines give good results. Random Forest Classifier is preferred because they have a more intuitive design and are easy-to-understand. And for ternary classification we can observe that Logistic regression gives good results.The experiments were conducted for four trials, each with five iterations (Itr) and the results are tabulated. We mentioned the average (Avg) of five iterations of each trial in the last column of each table for every technique.

6.1 Binary Sentiment Classification

The following are the accuracies where we considered only positive and negative polarities.

Trial-1

Table 1: Results of the Trial-1

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	78.55	75.93	80.12	75.60	78.65	77.77
L R	82.94	80.93	81.93	80.44	84.54	82.15
SVM	82.94	85.24	84.18	80.80	80.87	82.81
MLP	75.71	72.83	74.45	77.86	73.92	74.95
D T	70.80	73.30	71.79	68.95	72.44	71.45
R F	82.94	85.22	81.99	85.55	84.52	84.05

Trial-2

Table 2: Results of the Trial-2

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	78.55	80.42	79.96	78.91	79.20	79.41
L R	82.94	83.16	80.25	82.33	82.58	82.25
SVM	82.94	83.35	80.60	83.76	81.90	82.51
MLP	75.71	78.53	77.95	78.48	73.36	76.81
D T	68.47	66.30	71.26	69.81	68.56	68.88
R F	82.94	81.76	84.06	83.01	85.21	83.40

Trial-3

Table 3: Results of the Trial-3

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	78.55	76.10	79.05	80.02	80.10	78.77
L R	82.94	83.98	83.01	81.02	81.51	82.49
SVM	82.94	82.23	83.89	83.26	82.75	83.01
MLP	75.71	74.48	74.69	78.55	76.45	75.97
D T	71.05	72.55	70.27	71.68	73.84	71.88
R F	82.94	81.26	84.78	84.40	84.62	83.60

Trial-4

Table 4: Results of the Trial-4

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	78.55	76.47	75.58	76.30	78.78	77.14
L R	82.94	80.31	85.85	82.42	83.23	82.95
SVM	82.94	82.30	80.28	82.04	80.47	81.61
MLP	75.71	74.58	77.95	72.99	75.11	75.27
D T	68.99	68.28	70.57	68.21	67.12	68.64
R F	83.20	85.70	82.64	84.58	80.56	83.34

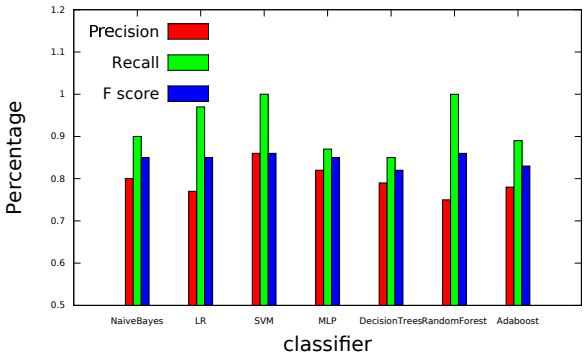
We used the ensemble of all the above six classifiers through a weighted majority vote to produce the final prediction. We get an accuracy of 73.85% for binary classification.

Precision, Recall and F-measure

Table 5: Precision, Recall and F-measure for Binary Classification

	Precision	Recall	F-measure
N B	0.80	0.90	0.85
L R	0.77	0.97	0.85
SVM	0.86	1.0	0.86
MLP	0.82	0.87	0.85
D T	0.79	0.85	0.82
R F	0.75	1.0	0.86
A B	0.78	0.89	0.83

Figure 4: Precision, Recall and F-measure for Binary classification



6.2 Ternary Sentiment Classification

There may be few cases where the data contains few sentences which may not contain any sentiment. So we considered neutral polarity leading to a ternary sentiment classification problem. The following are the accuracies where we considered all three polarities i.e., positive, negative and neutral polarities.

Trial-1

Table 6: Results of the Trial-1

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	64.54	65.1	64.85	67.42	61.96	64.77
L R	67.85	65.31	68.14	69.55	67.40	67.65
SVM	53.25	51.85	50.34	56.00	51.32	52.55
MLP	60.68	63.16	58.93	58.33	59.01	60.02
D T	52.42	54.94	50.18	54.18	52.89	52.92
R F	61.79	64.54	64.24	59.79	62.41	62.55

Trial-2

Table 7: Results of the Trial-2

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	64.54	65.66	63.85	63.59	63.37	64.20
L R	67.85	65.31	68.14	69.55	67.40	67.65
SVM	53.25	52.83	54.43	52.12	52.34	52.99
MLP	60.68	58.35	60.37	60.85	61.91	60.43
D T	54.90	51.96	55.01	52.26	53.15	53.45
R F	64.82	63.95	62.89	66.27	65.15	64.61

Trial-3

Table 8: Results of the Trial-3

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	64.54	67.17	61.80	63.78	65.54	64.56
L R	67.85	69.96	70.62	66.67	65.75	68.17
SVM	53.25	50.25	51.24	53.38	54.89	52.60
MLP	60.68	63.10	60.91	58.51	58.64	60.36
D T	51.87	50.72	50.53	51.86	51.20	51.23
R F	66.19	66.45	67.16	65.82	67.17	66.55

Trial-4

Table 9: Results of the Trial-4

	Itr 0	Itr 1	Itr 2	Itr 3	Itr 4	Avg
N B	64.54	61.86	67.20	64.18	66.48	64.85
L R	67.85	65.15	69.78	66.06	65.03	66.77
SVM	53.25	55.14	50.30	54.89	52.61	53.23
MLP	60.68	58.88	61.89	60.82	63.15	61.08
D T	51.87	52.19	53.76	54.59	52.85	53.05
R F	62.89	64.04	60.37	65.37	65.11	63.55

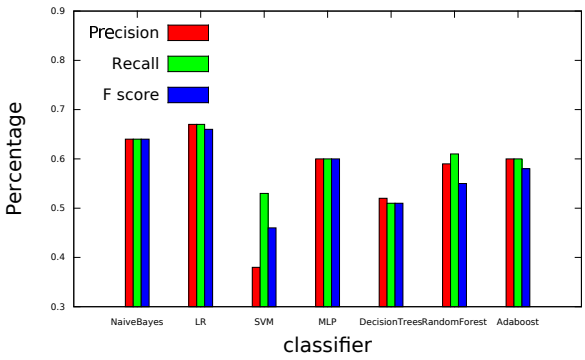
We used the ensemble of all the above six classifiers through a weighted majority vote to produce the final prediction. We get an accuracy of 60.13% for ternary classification.

Precision, Recall and F-measure

Table 10: Precision, Recall and F-measure for Ternary Classification

	Precision	Recall	F-measure
N B	0.64	0.64	0.64
L R	0.67	0.67	0.66
SVM	0.38	0.53	0.46
MLP	0.60	0.60	0.60
D T	0.52	0.51	0.51
R F	0.59	0.61	0.55
A B	0.60	0.60	0.58

Figure 5: Precision, Recall and F-measure for Ternary classification



7 Conclusion

Telugu is an agglutinative language. Considering this fact, we have achieved good results. Sentiment Analysis has not yet been tried on agglutinative Dravidian Languages. Since our work is the first attempt of this kind, we are not able to discuss comparative results. This approach produces a more focused and accurate sentiment summary of a given Telugu sentence which is useful for the users. This approach is not restricted by any domain. However, small modifications in the pre-processing would be sufficient to use this algorithmic formulation in different domains or languages.

Future Work

- To build a dictionary of frequently occurring positive and negative words and construct a lexicon-based system using it.
- To integrate a Morph Analyser to address the issue of agglutination.
- To test the system for different Indian languages.
- To work on the trending code-mixed data.

References

[Bakliwal *et al.*, 2011] Akshat Bakliwal, Piyush Arora, Ankit Patil, and V Verma. Towards enhanced opinion classification using nlp techniques. In *Proceedings of the 5th international joint conference on natural language processing (IJCNLP). Chiang Mai, Thailand*, pages 101–107. Citeseer, 2011.

[Balamurali, 2012] AR Balamurali. Cross-lingual sentiment analysis for indian languages using linked wordnets. 2012.

[Das and Bandyopadhyay, 2010] Amitava Das and Sivaji Bandyopadhyay. Sentiwordnet for indian languages. *Asian Federation for Natural Language Processing, China*, pages 56–63, 2010.

[Das and Bandyopadhyay, 2011] Amitava Das and Sivaji Bandyopadhyay. Dr sentiment knows everything! In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies: systems demonstrations*, pages 50–55. Association for Computational Linguistics, 2011.

[Das *et al.*,] Dipankar Das, Soujanya Poria, Chandra Mohan Dasari, and Sivaji Bandyopadhyay. Building resources for multilingual affect analysis–a case study on hindi, bengali and telugu. In *Workshop Programme*, page 54. Citeseer.

[Go *et al.*, 2009] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12, 2009.

[Harris, 1954] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[Joshi *et al.*, 2010] Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. A fall-back strategy for sentiment analysis in hindi: a case study. *Proceedings of the 8th ICON*, 2010.

[Le and Mikolov, 2014] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[Lee and Choeh, 2014] Sangjae Lee and Joon Yeon Choeh. Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications*, 41(6):3041–3046, 2014.

[Liu, 2010] Bing Liu. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666, 2010.

[Liu, 2012] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[Maas *et al.*, 2011] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[Mullen and Collier, 2004] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418, 2004.

[Narayanan *et al.*, 2013] Vivek Narayanan, Ishan Arora, and Arjun Bhatia. Fast and accurate sentiment classification using an enhanced naive bayes model. In *Intelligent Data Engineering and Automated Learning–IDEAL 2013*, pages 194–201. Springer, 2013.

[Pang and Lee, 2008] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[Patra *et al.*, 2015] Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Mining Intelligence and Knowledge Exploration*, pages 650–655. Springer, 2015.

[Wilson *et al.*, 2005] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.