#SarcasmDetection Is *Soooo* General! Towards a Domain-Independent Approach for Detecting Sarcasm

Natalie Parde and **Rodney D. Nielsen**
Department of Computer Science and Engineering
University of North Texas
natalie.parde, rodney.nielsen@unt.edu

Abstract

Automatic sarcasm detection methods have traditionally been designed for maximum performance on a specific domain. This poses challenges for those wishing to transfer those approaches to other existing or novel domains, which may be typified by very different language characteristics. We develop a general set of features and evaluate it under different training scenarios utilizing in-domain and/or out-of-domain training data. The best-performing scenario, training on both while employing a domain adaptation step, achieves an F_1 of 0.780, which is well above baseline F_1-measures of 0.515 and 0.345. We also show that the approach outperforms the best results from prior work on the same target domain.

Introduction

Sarcasm, a creative device used to communicate an intended meaning that is actually the opposite of its literal meaning,[1] is notoriously difficult to convey and interpret through text, in part because doing so relies heavily upon shared contextual understandings that can be marked more easily by altered prosody (e.g., emphasis upon certain words) or nonverbal signals (e.g., rolling one's eyes). It is a complex process even for humans, and in fact an inability to detect sarcasm has been linked with a number of neurocognitive disorders, including dementia (Kipps et al. 2009). It is similarly a challenging open task in natural language processing, and has direct implications to a number of other critical application areas, such as sentiment analysis.

Most research on automatic sarcasm detection to date has focused on the Twitter domain, which boasts an ample source of publicly-available data, some of which is already self-labeled by users for the presence of sarcasm (e.g., with #sarcasm). However, Twitter is highly informal, space-restricted, and subject to frequent topic fluctuations from one post to the next due to the ebb and flow of current events—in short, it is not broadly representative of most text domains. Thus, sarcasm detectors trained using features designed for maximum Twitter performance are not necessarily transferable to other domains. Despite this, it is desirable to develop approaches that can harness the more generalizable information present in the abundance of Twitter data.

In this work, we develop a set of domain-independent features for sarcasm detection and show that the features generally perform well across text domains. Further, we validate that domain adaptation can be applied to sarcasm detection to leverage patterns in out-of-domain training data, even when results from training only on that source domain data are extremely bad (far below baseline results), to improve over training on only the target data or over training on the simply combined dataset. Finally, we make a new dataset of sarcastic and non-sarcastic tweets available online as a resource to other researchers.[2]

Related Work

The majority of work on automatic sarcasm detection has been done using Twitter, and to a smaller extent Amazon product reviews. Research outside of those domains has been scarce, but interesting. Notably, Burfoot and Baldwin (2009) automatically detected satirical news articles using unigrams, lexical features, and semantic validity features, and Justo et al. (2014) used n-gram, linguistic, and semantic features to detect the presence of sarcasm in the Internet Argument Corpus (Walker et al. 2012). The remainder of this section describes prior work with Twitter and Amazon.

Sarcasm Detection on Twitter

Twitter is a micro-blogging service that allows users to post short "tweets" to share content or describe their feelings or opinions in 140 characters or less. For researchers, it boasts a low cost of annotation and plentiful supply of data (users often self-label their tweets using the "#" symbol—many explicitly label their sarcastic tweets using the hashtag "#sarcasm"). A variety of approaches have been taken toward automatically detecting sarcasm on Twitter, including explicitly using the information present in a tweet's hashtag(s); Maynard and Greenwood (2014) learned which hashtags characteristically corresponded with sarcastic tweets, and used the presence of those indicators to predict other sarcastic tweets, with high success. Liebrecht, Kunneman, and van den Bosch (2013) detected sarcasm in Dutch tweets using unigram, bigram, and trigram features.

Rajadesingan, Zafarani, and Liu (2015) detected sarcastic tweets based on features adapted from behavioral models of

[1] merriam-webster.com/dictionary/sarcasm

[2] hilt.cse.unt.edu/resources.html

sarcasm usage, drawing extensively from individual users' Twitter histories and relying heavily on situational context and user characteristics. The system also employed lexical features and grammatical correctness as a means of modelling different aspects of the user's behavior.

Other researchers have had success identifying sarcasm by a tweet's use of positive sentiment to describe a negative situation (Riloff et al. 2013), employing contextual (Bamman and Smith 2015) or pragmatic (González-Ibáñez, Muresan, and Wacholder 2011) features, and observing the writing style and emotional scenario of a tweet (Reyes, Rosso, and Veale 2013). An underlying theme among these methods is that the features are generally designed specifically for use with tweets. A major challenge in developing a more general approach for sarcasm detection lies in developing features that are present across many domains, yet still specific enough to reliably capture the differences between sarcastic and non-sarcastic text.

Finally, some researchers have recently explored approaches that rely on word embeddings and/or carefully tailored neural networks, rather than on task-specific feature design (Ghosh, Guo, and Muresan 2015; Ghosh and Veale 2016; Amir et al. 2016). Since neural networks offer little transparency, it is uncertain whether the features learned in these approaches would be easily transferable across text domains for this task (prior research on other tasks suggests that the features computed by deep neural networks grow increasingly specific to the training dataset—and in turn, to the training domain—with each layer (Yosinski et al. 2014)). Although an interesting question, the focus herein is on uncovering the specific types of features capable of leveraging general patterns for sarcasm detection, and this can be more easily examined using shallower learning algorithms.

Sarcasm Detection on Amazon Reviews

Research on automatic sarcasm detection in other domains has been limited, but recently a publicly-available corpus of sarcastic and non-sarcastic Amazon product reviews was released by Filatova (2012) to facilitate research. Buschmeier, Cimiano, and Klinger (2014) test many feature combinations on this dataset, including those based on metadata (e.g., Amazon star rating), sentiment, grammar, the presence of interjections (e.g., "wow") or laughter (e.g., through onomatopoeia or acronyms such as "lol"), the presence of emoticons, and bag-of-words features. Their highest F_1 (0.744) is achieved using all of these with a logistic regression classifier; however, using only the star rating, they still achieve an F_1 of 0.717. This highlights the need for high-performing, general features for sarcasm detection; metadata features are highly domain-specific, and even bag-of-words trends may be unique to certain domains ("trump" was one of the most common unigrams in our own Twitter training set, but only occurred once across all Amazon product reviews).

Prior to the release of Filatova's dataset, Davidov, Tsur, and Rappoport (2010) developed a semi-supervised approach to classify tweets or Amazon reviews as sarcastic or non-sarcastic by clustering samples based on grammatical features and the full or partial presence of automatically-extracted text patterns. They evaluated their work on a sample of the classified instances annotated by anonymous users on Amazon Mechanical Turk. They tested several different seed sets with their approach, one of which contained a mixture of positive Amazon reviews, positive #sarcasm-tagged tweets, and a manually-selected sample of negative tweets. Although they did not report test results on Amazon reviews using this seed set, they did report test results on #sarcasm-tagged tweets, achieving an F-measure of 0.545. Their work is the closest to ours, because it attempts to harness training samples from both the Twitter and Amazon review domains.

	Sarcastic (Train/Test)	Non-Sar. (Train/Test)
Twitter	1959 (1568/391)	3039 (2430/609)
Amazon	437 (350/87)	817 (653/164)

Table 1: Twitter and Amazon Dataset Distributions

Methods

Data Collection

Data was taken from two domains: Twitter, and Amazon product reviews. The Amazon reviews were from the publicly available sarcasm corpus developed by Filatova (2012). To build our Twitter dataset, tweets containing exactly one of the trailing hashtags "#sarcasm," "#happiness," "#sadness," "#anger," "#surprise," "#fear," and "#disgust" were downloaded regularly during February and March 2016. Tweets containing the latter six hashtags, corresponding to Ekman's six basic emotions (Ekman 1992), were labeled as non-sarcastic. Those hashtags were chosen because their associated tweets were expected to still express opinions, similarly to sarcastic tweets, but in a non-sarcastic way. Tweets containing #sarcasm were labeled as sarcastic; annotating tweets with the #sarcasm hashtag as such is consistent with the vast majority of prior work in the Twitter domain (González-Ibáñez, Muresan, and Wacholder 2011; Liebrecht, Kunneman, and van den Bosch 2013; Maynard and Greenwood 2014; Rajadesingan, Zafarani, and Liu 2015; Bamman and Smith 2015; Ghosh, Guo, and Muresan 2015; Amir et al. 2016).

The downloaded tweets were filtered to remove retweets, "@replies," and tweets containing links. Retweets were removed to avoid having duplicate copies of identical tweets in the dataset, @replies were removed in case the hashtag referred to content in the tweet to which it replied rather than content in the tweet itself, and tweets with links were likewise removed in case the hashtag referred to content in the link rather than in the tweet itself. Requiring that the specified hashtag trailed the rest of the tweet (it could only be followed by other hashtags) was done based on the observation that when sarcastic or emotional hashtags occur in the main tweet body, the tweet generally discusses sarcasm or the specified emotion, rather than actually expressing sarcasm or the specified emotion. Finally, requiring that only one of the specified hashtags trailed the tweet eliminated cases of ambiguity between sarcastic and non-sarcastic tweets. All trailing "#sarcasm" or emotion hashtags were removed from the data before training and testing, and both datasets were

Resource	Description
Liu05	Opinion lexicon containing 2006 pos. words and 4783 neg. words (Liu, Hu, and Cheng 2005).
MPQA	Subjectivity lexicon containing strongly or weakly subjective positive (2718) or negative (4910) words (Wilson, Wiebe, and Hoffmann 2005).
AFINN	Sentiment lexicon for microblogs (Hansen et al. 2011) containing 2477 words/phrases labeled with values from -5 (negative) to +5 (positive).
Google Web1T	Large collection of n-grams and their frequencies scraped from the web (Brants and Franz 2006).

Table 2: Lexical Resources

randomly divided into training (80%) and testing (20%) sets. Further details are shown in Table 1.

Features

Three feature sets were developed (one general, and two targeted toward Twitter and Amazon, respectively). Resources used to develop the features are described in Table 2. Five classifiers (Naïve Bayes, J48, Bagging, DecisionTable, and SVM), all from the Weka[3] library, were tested using five-fold cross-validation on the training sets, and the highest-scoring (Naïve Bayes) was selected for use on the test set.

Domain-Specific Features The Twitter- (*T*) and Amazon-specific (*A*) features are shown in Table 3. Domain-specific features were still computed for instances from the other domain unless it was impossible to compute those features in that domain (i.e., Amazon Star Rating for Twitter instances), in which case they were left empty. Twitter-specific features are based on the work of Maynard and Greenwood (2014) and Riloff et al. (2013). Maynard and Greenwood detect sarcastic tweets by checking for the presence of learned hashtags that correspond with sarcastic tweets, as well as sarcasm-indicator phrases and emoticons. We construct binary features based on their work, and on Riloff et al.'s work (2013), which determined whether or not a tweet was sarcastic by checking for positive sentiment phrases contrasting with negative situations (both of which were learned from other sarcastic tweets). We also add a feature indicating the presence of laughter terms. Amazon-based features are primarily borrowed from Buschmeier, Cimiano, and Klinger's (2014) earlier work on the Amazon dataset.

Feature	Description
Contains Sarcasm Hashtag	*(T)* True if contains hashtag learned by Maynard & Greenwood (excluding #sarcasm).[4]
Contains Sarcastic Smiley	*(T)* True if instance contains a sarcastic emoticon learned by Maynard & Greenwood.
Contains Sar. Indicator	*(T)* True if instance contains a sarcasm indicator phrase learned by Maynard & Greenwood.
Contains Positive Predicate	*(T)* True if instance contains a positive predicate learned from Twitter by Riloff.
Contains Pos. Sentiment	*(T)* True if instance contains a positive sentiment phrase learned from Twitter by Riloff.
Contains Neg. Situation	*(T)* True if instance contains a negative situation phrase learned from Twitter by Riloff.
Pos. Sent. Precedes Neg. Situation	*(T)* True if contains a pos. predicate or sentiment phrase learned by Riloff that precedes a learned neg. situation phrase by $\leq$ 5 tokens.
Contains Laughter	*(T)* True if instance contains *hahaha*, *haha*, *hehehe*, *hehe*, *jajaja*, *jaja*, *lol*, *lmao*, or *rofl*.[4]
Star Rating	*(A)* Numeric score (1-5) corresponding to number of stars associated with the review.
Contains *Wow*	*(A)* True if the instance contains *wow*.
Contains *Ugh*	*(A)* True if the instance contains *ugh*.
Contains *Huh*	*(A)* True if the instance contains *huh*.
Contains "..."	*(A)* True if the instance contains an ellipsis.

Table 3: Domain-Specific Features

General Features We model some of our general features after those from Riloff et al. (2013), under the premise that the underlying principle that sarcasm often associates positive expressions with negative situations holds true across domains. Since positive sentiment phrases and negative situations learned from tweets are unlikely to generalize to different domains, we instead use three sentiment lexicons to build features that capture positive and negative sentiment rather than checking for specific learned phrases. Likewise, rather than bootstrapping specific negative situations from Twitter, we calculate the pointwise mutual information (PMI) between the most positive or negative word in the instance and the n-grams that immediately proceed it[5] to create a more general version of the feature. Other general features developed for this work rely on syntactic characteristics, or are bag-of-words-style features corresponding to the tokens most strongly correlated or most common in sarcastic and non-sarcastic instances from Twitter and Amazon training data. All general features are outlined in Table 4.

Evaluation

The features used for each train/test scenario are shown in the first column of Table 5. *Twitter Features* refers to all features listed in Table 3 preceded by the parenthetical *(T)*, and *Amazon Features* to all features preceded by *(A)*. *General: Other Polarity* includes the positive and negative percentages, average polarities, overall polarities, and largest polarity gap features from Table 4. *General: Subjectivity* includes the % strongly subjective positive words, % weakly subjective positive words, and their negative counterparts. We also include two baselines: the *All Sarcasm* case assumes that every instance is sarcastic, and the *Random* case randomly assigns each instance as sarcastic or non-sarcastic.

Results are reported for models trained only on Twitter, only on Amazon, on both training sets, and on both training sets when Daumé's (2007) EasyAdapt technique is applied, employing Twitter as the algorithm's source domain and Amazon as its target domain. EasyAdapt works by modifying the feature space so that it contains three mappings of the original features: a general (source + target) version, a

[3] cs.waikato.ac.nz/ml/weka/

[4] Individual binary features for each of the sarcasm hashtags (5 features) and laughter tokens (9 features) were also included.

[5] Frequencies for computing PMI were from Google Web1T.

Feature	Description
Most Polar Unigram	The single most positive or negative unigram in the instance, according to AFINN.
Most Polar Score	The score, ranging from -5 to +5, corresponding to the most polar unigram.
% Strongly (Weakly) Subj. Pos. (Neg.) Words	Four features: The number of words identified as strongly (weakly) subjective positive (negative) words in the instance according to MPQA, divided by the count of the instance's words found in MPQA.
Avg. Polarity of the Instance (Liu05)	The sum of the polarity scores for all words (positive words = $+1$, negative words = -1) in the instance included in Liu05, divided by the total number of words in the instance included in Liu05.
Avg. Polarity of the Instance (MPQA)	An analogue of the above, for MPQA. Strongly subjective pos. words are assigned a score of +2, weakly subjective pos. words +1, weakly subjective neg. words -1, and strongly subjective neg. words -2.[6]
Avg. Polarity of the Instance (AFINN)	The sum of the polarity scores for all words in the instance included in the AFINN sentiment lexicon, divided by the total number of words in the instance included in that lexicon.
Overall Polarity of the Instance	Three separate scores, corresponding to the sum of the polarity scores for all words in the Liu05, MPQA, and AFINN lexicons, respectively (scores are calculated as described above).
% Positive (Negative) Words	Six separate features, calculated by dividing the number of positive (or negative) words in the instance according to Liu05, MPQA, and AFINN, respectively, by the total number of words in the instance.
N-gram PMI Scores	Four features corresponding to the PMI between the most polar unigram and the 1-, 2-, 3-, and 4-grams that immediately follow it. Let $\mathrm{p}(w, W_n)$ be the probability of the sequence starting with unigram w and ending with the n-gram W_n of size n, where $\mathrm{C}(w, W_n)$ is the number of occurrences of w immediately followed by W_n, and N is the count of all n-grams of length $n+1$ in the corpus. Then, $\mathrm{p}(w, W_n) = \frac{1}{N}\mathrm{C}(w, W_n), \qquad \mathrm{PMI}(w, W_n) = \log \frac{\mathrm{p}(w, W_n)}{\mathrm{p}(w, *_n) \times \mathrm{p}(*, W_n)} \qquad (1)$ where $*_n$ can be any n-gram of length n and $*$ can be any unigram. In tweets, hashtags are removed prior to calculating PMI (e.g., "#love" becomes "love"), and any tokens beginning with "@" may be matched by any token (these are assumed to be mentions of another Twitter user by his or her username).
All-Caps Words	Two features, corresponding to the raw number of all-caps words in the instance, and the number of all-caps words divided by the total number of words in the instance.
Consecutive Chars.	The highest number of consecutive repeated characters in the instance (e.g., "Sooooo" $\Rightarrow 5$).
Consecutive Punct.	The highest number of consecutive punctuation characters in the instance.
Specific Character Features	Two binary features: one is equal to 1 if the instance contains an exclamation mark, and the other is equal to 1 if the instance contains a question mark.
Largest Score Gap	The most negative score in the instance, according to the AFINN lexicon, subtracted from the most positive score in the instance, according to the AFINN lexicon.
Bag-of-Associated-Words (BOAW)	Up to 200 features: training instances were divided into four groups (Sarcastic × Non-Sarcastic) × (Amazon × Twitter). For each group, the 50 unigrams most strongly correlated with that class and domain were computed based on the PMI between the unigram and class label. Specifically: $\mathrm{PMI}(w, l) = \log \frac{\mathrm{p}(w,l)}{\mathrm{p}(w) \times \mathrm{p}(l)}$, where w is the unigram, l is a label, $\mathrm{p}(w, l)$ is the joint probability of an instance containing w and being labeled l, $\mathrm{p}(w)$ is the probability of w being in any instance, and $\mathrm{p}(l)$ is the fraction of instances labeled l. Probabilities were computed separately for Amazon and Twitter; stopwords are removed prior to calculating PMI. Plus-one smoothing was used for all probabilities.
Bag-of-Common-Words (BOCW)	Up to 200 features: training instances were divided into the same four groups as above. For each group the 50 most common unigrams were determined. Any duplicates across groups were then removed.

Table 4: General Feature Set

source-only version, and a target-only version. More specifically, assuming an input feature set $\mathcal{X} = \mathbb{R}^F$ for some $F > 0$, where F is the number of features in the set, EasyAdapt transforms $\mathcal{X}$ to the augmented set, $\check{\mathcal{X}} = \mathbb{R}^{3F}$. The mappings $\Phi^s, \Phi^t : \mathcal{X} \rightarrow \check{\mathcal{X}}$ for the source and target domain data, respectively, are defined as $\Phi^s(\mathbf{x}) = \langle \mathbf{x}, \mathbf{x}, \mathbf{0} \rangle$ and $\Phi^t(\mathbf{x}) = \langle \mathbf{x}, \mathbf{0}, \mathbf{x} \rangle$, where $\mathbf{0} = \langle 0, ..., 0 \rangle \in \mathbb{R}^F$ is the zero vector. Refer to Daumé (2007) for an in-depth discussion of this technique.

Each model was tested on the Amazon test data (the model trained only on Twitter was also tested on the Twitter test set). Amazon reviews were selected as the target domain since the Twitter dataset was much larger than the Amazon dataset; this scenario is more consistent with the typically stated goal of domain adaptation (a large labeled out-of-domain source dataset and a small amount of labeled data in the target domain), and most clearly highlights the need for a domain-general approach.

Finally, we include the best results reported by Buschmeier, Cimiano, and Klinger (2014) on the same Amazon dataset. For a more direct comparison between our work and theirs, we also report the results from using all of our features under the same classification conditions as theirs (10-fold cross-validation using *scikit-learn*'s Logistic Regression,[7] tuning with an F_1 objective). We refer to the latter case as *Our Results, Same Classifier as Prior Best*.

Results

The results, including each of the training scenarios noted earlier, are presented in Table 5. Precision, recall, and F_1 on the positive (sarcastic) class were recorded. The highest F_1 achieved (0.780) among all cases was from training on the EasyAdapted Twitter and Amazon data. In comparison, training only on the Amazon reviews produced an F_1 of 0.713 (training and testing only on Amazon reviews with our features but with the same classifier and cross-validation

[6] Part-of-speech is considered in MPQA; Amazon and Twitter data was tagged using Stanford CoreNLP (Manning et al. 2014) and the Twitter POS-tagger (Owoputi et al. 2013), respectively.

[7] scikit-learn.org

	Test on Amazon Reviews												Test on Twitter		
	Train on Twitter			Train on Both			Train on Amazon			EasyAdapt			Train on Twitter		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Baseline: All Sarcasm	.35	1.0	.515	.35	1.0	.515	.35	1.0	.515	.35	1.0	.515	.39	1.0	.562
Baseline: Random	.35	.35	.347	.35	.35	.347	.35	.35	.347	.35	.35	.347	.39	.39	.391
Twitter Features	.32	.36	.337	.00	.00	.000	.00	.00	.000	.00	.00	.000	.57	.24	.341
Amazon Features	.48	.26	.341	.76	.68	.715	.76	.67	.712	.76	.68	.715	.47	.14	.216
Gen.: Most Polar Word	.26	.30	.275	.43	.15	.222	.68	.20	.304	.52	.33	.406	.65	.38	.479
Gen.: Most Polar Score	.26	.48	.335	.00	.00	.000	.59	.46	.516	.59	.46	.516	.56	.40	.466
General: Other Polarity	.26	.36	.302	.35	.40	.374	.61	.69	.645	.55	.67	.601	.51	.58	.542
General: Subjectivity	.25	.02	.042	.25	.02	.042	.60	.36	.446	.59	.36	.443	.48	.15	.229
General: Syntactic	.51	.26	.348	.69	.36	.470	.71	.43	.532	.65	.51	.568	.46	.17	.250
General: PMI Features	.00	.00	.000	.25	.03	.061	.34	.14	.197	.42	.21	.277	.60	.02	.030
General: BOAW	.00	.00	.000	.00	.00	.000	.63	.20	.298	.60	.23	.331	.00	.00	.000
General: BOCW	.47	.59	.523	.58	.16	.252	.59	.60	.594	.63	.46	.530	.59	.39	.467
All General Features	.26	.32	.290	.42	.29	.340	.63	.69	.659	.69	.67	.678	.55	.62	.582
All Features	.25	.31	.276	.66	.54	.595	.66	.77	.713	.75	.82	**.780**	.55	.62	.583
Prior Best Results (Buschmeier, Cimiano, and Klinger 2014)							.82	.69	.744						
Our Results, Same Classifier as Prior Best							.80	.71	**.752**						

Table 5: Test Results — Full Analysis

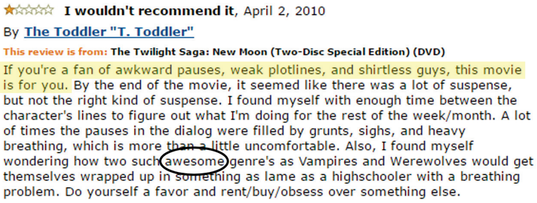
settings as Buschmeier, Cimiano, and Klinger (2014) led to an F_1 of 0.752, outperforming prior best results on that dataset). Training on both without EasyAdapt led to an F_1 of 0.595 (or 0.715 when training only on Amazon-specific features), and finally, training only on Twitter data led to an F_1 of 0.276. Training and testing on Twitter produced an F_1 of 0.583 when training on all features.[8]

Discussion

When testing on Amazon reviews, the worst-performing case was that in which the classifier was trained only on Twitter data (it did not manage to outperform either baseline). This underscores the inherent variations in the data across the two domains; despite the fact that many of the features were deliberately designed to be generalizable and robust to domain-specific idiosyncrasies, the different trends across domains still confused the classifier.

However, combining all of that same Twitter data with a much smaller amount of Amazon data (3998 Twitter training instances relative to 1003 Amazon training instances) and applying EasyAdapt to the combined dataset performed quite well (F_1=0.780). The classifier was able to take advantage of a wealth of additional Twitter samples that had led to terrible performance on their own (F_1=0.276). Thus, the high performance demonstrated when the EasyAdapt algorithm is applied to the training data from the two domains is particularly impressive. It shows that more data is indeed better data—provided that the proper features are selected and the classifier is properly guided in handling it.

Overall, the system cut the error rate from .256 to .220, representing a 14% relative reduction in error over prior best results on the Amazon dataset. Our results testing on Twitter are not directly comparable to others, since prior work's datasets could not be released; however, our results (F_1=0.583) are in line with those reported previously (Riloff et al. (2013): F_1=0.51; Davidov, Tsur, and Rappoport (2010): F_1=0.545). Additionally, our Twitter data did not contain many indicators shown to be discriminative in the past (leading our general features to be better predictors of sarcasm even when training/testing entirely within the domain), and our focus in developing features was on general performance rather than performance on Twitter specifically.



★☆☆☆☆ **I wouldn't recommend it**, April 2, 2010

By **The Toddler "T. Toddler"**

This review is from: The Twilight Saga: New Moon (Two-Disc Special Edition) (DVD)

If you're a fan of awkward pauses, weak plotlines, and shirtless guys, this movie is for you. By the end of the movie, it seemed like there was a lot of suspense, but not the right kind of suspense. I found myself with enough time between the character's lines to figure out what I'm doing for the rest of the week/month. A lot of times the pauses in the dialog were filled by grunts, sighs, and heavy breathing, which is more than a little uncomfortable. Also, I found myself wondering how two such awesome genre's as Vampires and Werewolves would get themselves wrapped up in something as lame as a highschooler with a breathing problem. Do yourself a favor and rent/buy/obsess over something else.

Figure 1: Example from Amazon Product Reviews

Both datasets were somewhat noisy. Many full-length reviews that were marked as "sarcastic" were only partially so, and included other sentences that were not sarcastic at all. This may have been particularly problematic when strong polarity was present in those sentences. An example of this is shown in Figure 1, where the highlighted portion of the review indicates the sarcastic segment submitted by the annotator, and *awesome*, the most polar word in the entire review (circled), is outside that highlighted sentence.

Since tweets are self-labeled, users' own varying definitions of sarcasm lead to some extreme idiosyncrasies in the kinds of tweets labeled as sarcastic. Sarcastic tweets were also often dependent upon outside context. Some examples include (#sarcasm tags were removed in the actual dataset): *"My daughter's 5th grade play went over as professional, flawless, and well rehearsed as a Trump speech. #sarcasm," "#MilanAlessandria Mario Balotelli scored the fifth goal in the 5-0 win. He should play for the #Azzurri at #EURO2016.*

[8] Further analysis of the Twitter-specific features revealed that contains_sarcasm_hashtag, contains_sarcastic_smiley, and contains_sarcasm_indicator_phrase all led to $F_1 = 0.0$ when used individually; although these performed quite well in prior work, our Twitter dataset did not contain the indicators with high enough frequency to have any impact on the overall classification outcome.

#sarcasm," and *"Good morning #sarcasm."*

Finally, some past research has found that it is more difficult to discriminate between sarcastic and non-sarcastic texts when the non-sarcastic texts contain sentiment (González-Ibáñez, Muresan, and Wacholder 2011; Ghosh, Guo, and Muresan 2015). Since our non-sarcastic tweets are emotionally-charged, our classifier may have exhibited lower performance than it would have with only neutral non-sarcastic tweets. Since distinguishing between literal and sarcastic sentiment is useful for real-world applications of sarcasm detection, we consider the presence of sentiment in our dataset to be a worthwhile challenge.

Regarding the general features developed for this work, the polarity- and subjectivity-based features performed well, while performance using only PMI features was lower. PMI scores in particular may have been negatively impacted by common Twitter characteristics, such as the trend to join keywords together in hashtags, and the use of acronyms that are unconventional in other domains. These issues could be addressed to some extent in the future via word segmentation tools, spell-checkers, and acronym expansion.

Conclusions

This work develops a set of domain-independent features and demonstrates their usefulness for general sarcasm detection. Moreover, it shows that by applying a domain adaptation step to the extracted features, even a surplus of "bad" training data can be used to improve the performance of the classifier on target domain data, reducing error by 14% relative to prior work. The Twitter corpus described in this paper is publicly available for research purposes,[2] and represents a substantial contribution to multiple NLP sub-communities. This shared corpus of tweets annotated for sarcasm will hasten the advancement of further research. In the future, we plan to extend our approach to detect sarcasm in a completely novel domain, literature, eventually integrating the work into an application to support reading comprehension.

Acknowledgments

This material is based upon work supported by the NSF Graduate Research Fellowship Program under Grant 1144248, and the NSF under Grant 1262860. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Amir, S.; Wallace, B. C.; Lyu, H.; Carvalho, P.; and Silva, M. J. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proc. of CoNLL.*

Bamman, D., and Smith, N. 2015. Contextualized sarcasm detection on twitter. In *Proc. of ICWSM*.

Brants, T., and Franz, A. 2006. Web 1t 5-gram v1.

Burfoot, C., and Baldwin, T. 2009. Automatic satire detection: Are you having a laugh? In *Proc. of ACL-IJCNLP*.

Buschmeier, K.; Cimiano, P.; and Klinger, R. 2014. An impact analysis of features in a classification approach to irony detection in product reviews. In *Proc. of WASSA*.

Daumé III, H. 2007. Frustratingly easy domain adaptation. In *Proc. of ACL*.

Davidov, D.; Tsur, O.; and Rappoport, A. 2010. Semi-supervised recognition of sarcasm in twitter and amazon. In *Proc. of CoNLL.*

Ekman, P. 1992. An argument for basic emotions. *Cognition and Emotion* 6(3-4).

Filatova, E. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proc. of LREC*.

Ghosh, A., and Veale, T. 2016. Fracking sarcasm using neural network. In *Proc. of WASSA*.

Ghosh, D.; Guo, W.; and Muresan, S. 2015. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proc. of EMNLP*.

González-Ibáñez, R.; Muresan, S.; and Wacholder, N. 2011. Identifying sarcasm in twitter: A closer look. In *Proc. of ACL-HLT*.

Hansen, L. K.; Arvidsson, A.; Nielsen, F. A.; Colleoni, E.; and Etter, M. 2011. Good friends, bad news - affect and virality in twitter. In *Proc. of FutureTech*.

Justo, R.; Corcoran, T.; Lukin, S.; Walker, M.; and Torres, M. 2014. Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web. *Knowl.-Based Sys.*

Kipps, C.; Nestor, P.; Acosta-Cabronero, J.; Arnold, R.; and Hodges, J. 2009. Understanding social dysfunction in the behavioural variant of frontotemporal dementia: the role of emotion and sarcasm processing. *Brain* 132(3).

Liebrecht, C.; Kunneman, F.; and van den Bosch, A. 2013. The perfect solution for detecting sarcasm in tweets# not. In *Proc. of WASSA*.

Liu, B.; Hu, M.; and Cheng, J. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *WWW'05*.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL Sys. Demos*.

Maynard, D., and Greenwood, M. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proc. of LREC*.

Owoputi, O.; Dyer, C.; Gimpel, K.; Schneider, N.; and Smith, N. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proc. of NAACL*.

Rajadesingan, A.; Zafarani, R.; and Liu, H. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proc. of WSDM*.

Reyes, A.; Rosso, P.; and Veale, T. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation* 47(1).

Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; and Huang, R. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proc. of EMNLP*.

Walker, M.; Tree, J. F.; Anand, P.; Abbott, R.; and King, J. 2012. A corpus for research on deliberation and debate. In *Proc. of LREC*.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of HLT-EMNLP*.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Proc. of NIPS*.