Disambiguating Sentiment: An Ensemble of Humour, Sarcasm, and Hate Speech Features for Sentiment Classification

Rohan Badlani [*]
Dept. of Computer Science
Stanford University
rbadlani@stanford.edu

Nishit Asnani [*]
Dept. of Computer Science
Stanford University
nasnani@stanford.edu

Manan Rai [*]
Dept. of Computer Science
Stanford University
mananrai@stanford.edu

Abstract

Due to the nature of online user reviews, sentiment analysis on such data requires a deep semantic understanding of the text. Many online reviews are sarcastic, humorous, or hateful. Signals from such language nuances may reinforce or completely alter the sentiment of a review as predicted by a machine learning model that attempts to detect sentiment alone. Thus, having a model that is explicitly aware of these features should help it perform better on reviews that are characterized by them. We propose a composite two-step model that extracts features pertaining to sarcasm, humour, hate speech, as well as sentiment, in the first step, feeding them in conjunction to inform sentiment classification in the second step. We show that this multi-step approach leads to a better empirical performance for sentiment classification than a model that predicts sentiment alone. A qualitative analysis reveals that the conjunctive approach can better capture the nuances of sentiment as expressed in online reviews.

1 Introduction

Sentiment classification is one of the most widely studied problems in natural language processing, partly since it is a complex problem from a linguistic point of view, and partly since it has huge commercial value for enterprises attempting to understand user behaviour. Online user review datasets have contributed significantly to research in this direction, since they provide large sets of human generated commentary about real products and services, which capture the nuances and complexity of the user-generated text (Zhang et al., 2015) (He and McAuley, 2016).

Traditionally, models developed for sentiment classification have been used to solve other binary or multi-class classification problems in natural language, like intent detection, document tagging, etc. Sentiment classification has also been used as a building block towards more complicated language understanding/generation tasks (Poria et al., 2016) (Hu et al., 2017). Given that sentiment is a complex language attribute influenced by several other features, this paper poses a question more fundamental to the nature of sentiment in human language: can models developed for other tasks, like sarcasm, humor, or hate speech detection, help improve sentiment classification models? Going further, we also attempt to answer if the same model architecture can be used for these tasks, and then be combined to yield higher performance on sentiment classification.

This line of thought is inspired by how humans perceive sentiment in any piece of spoken or written language. Detection of elements of sarcasm help us resolve seemingly contradictory statements ("The restaurant was so clean that I could barely avoid stepping into the puddle!") into their intended sentiment. Similarly, humor (because it can get similarly confusing) and hate speech (since specific offensive words may be the only indicators of sentiment in a review) act as crucial indicators of the intended meaning of phrases in a given utterance.

Since the sentiment model is not optimizing for detecting these language attributes, it is likely to get confused by utterances having them unless it is sufficiently exposed to similar sentences during training. We therefore believe that making sentiment models explicitly aware of these language attributes would help them become more robust to sarcastic, humorous or hateful utterances, and thus get better at classifying sentiment.

Thus, our research hypotheses are as follows:

[*]equal contribution

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 337–345
Hong Kong, Nov 4, 2019.

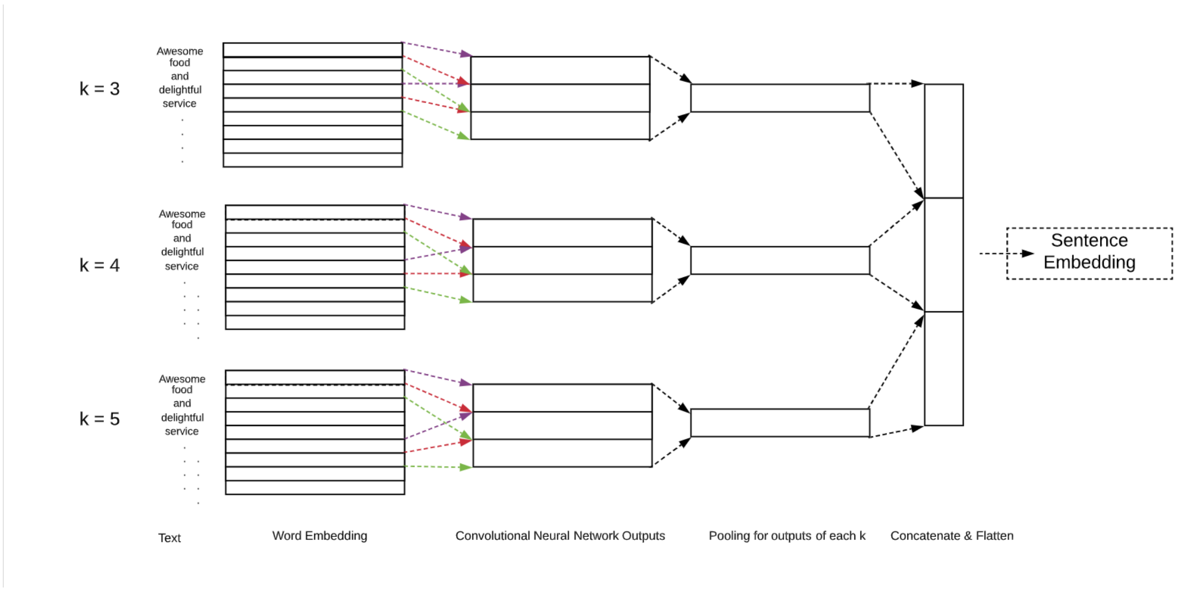

k = 3
k = 4
k = 5
Awesome food and delightful service
Sentence Embedding
Text
Word Embedding
Convolutional Neural Network Outputs
Pooling for outputs of each k
Concatenate & Flatten

Figure 1: CNN Based Binary Classification Model for embedding generation. We use a stride of 1 in our final CNN model. Different colors in the word embeddings represent different inputs to the convolutional neural network

- **H1**: Models individually learned on sarcasm, humor and hate speech detection, and then used as subroutines to extract features, should boost the performance of a sentiment classification model.

- **H2**: Given that the individual tasks are all binary classification tasks, we believe that a single model architecture should provide reasonable performance on these individual tasks and would make it easier to re-use the same learned models for multiple downstream classification tasks.

2 Related Work

Sentiment classification, sarcasm detection, humor detection and hate speech detection have all seen varying levels of interest from the natural language research community, and have evolved over time as better datasets and modeling techniques have come into the picture.

There has been quite a bit of work on sarcasm detection, especially in the context of Twitter-based self-annotated data and Self-Annotated Reddit Corpus. The seminal work in this area started with (González-Ibáñez et al., 2011) - they used lexical and pragmatic features and found that pragmatic features were more useful in detecting sarcasm. Addition of context-based features along with text-based features in certain subsequent models helped as well in improving performance on sarcasm detection. There was a dramatic shift with the introduction of deep learning as feature engineering took a back seat and deep models began to be used for learning task-specific representations. (Hazarika et al., 2018) show that using context, user and text embedding provides state of the art performance, which is challenged by Kolchinski (Kolchinski and Potts, 2018) (West et al., 2014) through a more simplistic user embedding based approach that achieves similar performance without other context (like forum embeddings as used by (Hazarika et al., 2018)).

Hate Speech in natural language research has traditionally been a loosely-defined term, with one cause being the similarity with other categorizations of hateful utterances, such as offensive language. In the context of online reviews, we broadly use hate speech to include any form of offensive language. (Davidson et al., 2017) introduce the seminal dataset in the field, and test a variety of models – Logistic Regression, Naive Bayes, decision trees, random forests, and Support Vector Machines (SVMs), each tested with 5-fold cross validation to find that the Logistic Regression and Linear SVM tend to perform significantly better than other models. Models such as LSTMs and CNNs have also been tried in works such as (Badjatiya et al., 2017) and (de Gibert et al., 2018).

Humour Detection has seen a lot of work, with models being developed on several large-scale public datasets, such as the Pun of the Day, 16000

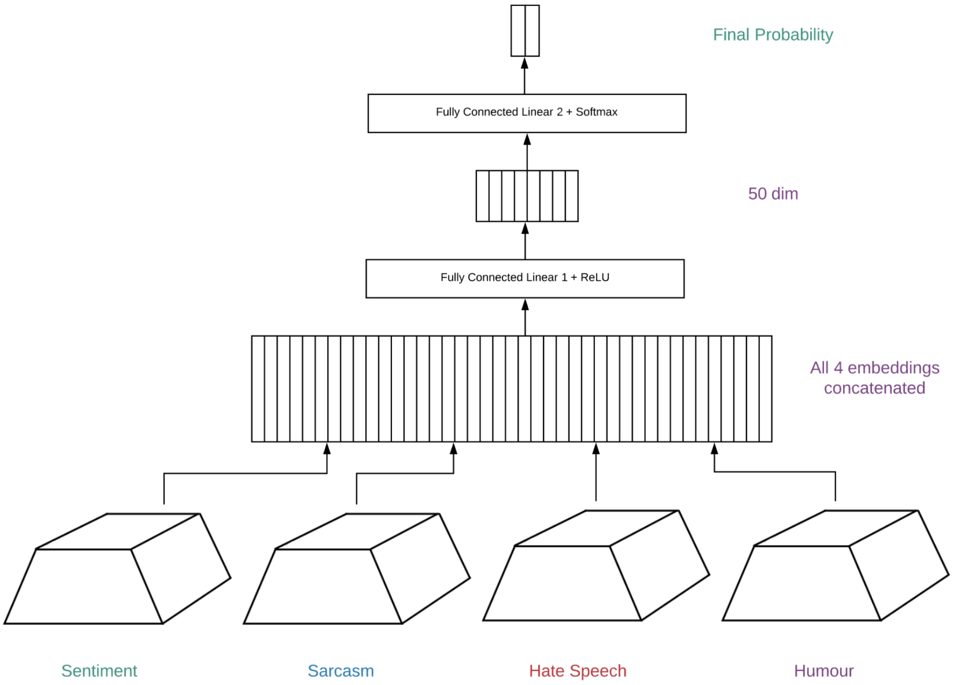

Final Probability
Fully Connected Linear 2 + Softmax
50 dim
Fully Connected Linear 1 + ReLU
All 4 embeddings concatenated
Sentiment
Sarcasm
Hate Speech
Humour

Figure 2: Full Sentiment Classification model with ensemble of features from sarcasm, humour, and hate speech detection models

OneLiners, Short Jokes dataset, and the PTT jokes dataset. (Chen and Soo, 2018) use a Highway Network on top of a CNN on a combination of these datasets. (Kim, 2014) uses CNN for sentence classification, and these models have also been tested on funny-labelled reviews from the Yelp dataset[1].

Recent works have attempted to combine feature extraction models trained on some tasks for a different task. (Poria et al., 2016), for instance, uses knowledge about sentiment, emotion, and personality to better detect sarcasm. This finds a parallel in our attempt here, with the difference that these features include non-linguistic features such as user personality, and we focus only on natural language features to test the transferability of knowledge about certain features to detecting others.

Sentiment classification is a text classification task with the objective to classify text according to the sentimental polarities. This has been a widely researched area (Mäntylä et al., 2016) and recently there has been a lot of success in this area. The current state of the art performance on this task is using transformer (Vaswani et al., 2017) based models like BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019). These models achieve very high accuracy on the binary classification task of sentiment polarity detection but analyzing the failure modes of these models indicate that these models might fail in cases where there are higher order language concepts like sarcasm, humour and hate speech co-occur in the same utterance. Hence, through this paper, we investigate the performance of sentiment classification when provided with representative features pertaining to these language oriented concepts and at the same time propose a generic approach to compute these feature so as to reuse for multiple downstream tasks.

3 Methods

3.1 Datasets

We experiment with the following datasets corresponding to sentiment, sarcasm, hate speech, and humour to test our hypotheses:

1. **D1 - Sentiment**: The Yelp Review Dataset (Zhang et al., 2015) consists of about 560,000 reviews, with binary sentiment labels. For the purposes of our analysis, we use 100,000 reviews for training and validation and 36,614

[1] https://github.com/srishti-1795/Humour-Detection

Review	**NLU Aspect Captured**
I am a nurse and to characterize this office as nothing but a frustration is a compliment.	Sarcasm
It would have been faster if I would have grown, harvested, and brewed the tea myself.	Sarcasm
If you want to get pastries while being yelled at by the staff and treated like dirt, this is the place for you.	Sarcasm
They are THE RUDEST people I've ever met! The lady with short hair has a crappy attitude, so does the younger guy.	Hate Speech

Table 1: Examples where our combined model is able to predict correct label whereas the baseline sentiment model fails.

reviews for testing.

2. **D2 - Sarcasm**: The SARC (Self-Annotated Reddit Comments) dataset (Khodak et al., 2017) consists of about 1.3 million Reddit comments. These comments have been self annotated using the \s character. The dataset has a balanced set and an unbalanced set. For the purposes of our analysis, we focus on balanced set and take 100,000 comments for training and validation and 20,000 comments for testing.

3. **D3 - Humour**: The Yelp Review Dataset has a field called '*funny*.' We consider a comment to be humorous (i.e positive label) when the comment has a 'funny' score greater than 2.

4. **D4 - Hate Speech**: The hatebase.org Twitter Dataset (Davidson et al., 2017) is a popular hate speech tweet dataset, which consists of 28,000 tweets, each labelled as either having offensive content or not.

3.2 Validating Hypothesis H1

We conduct a quick initial evaluation of hypothesis **H1** using well-performing models for sentiment, sarcasm, humour, and hate speech. These models are discussed below:

1. **M1 - Sentiment**: For sentiment detection, the current state of the art model is BERT Large (Devlin et al., 2018) which provides an accuracy of about 98.11% (Xie et al., 2019). We use the BERT Base model which has a smaller architecture and therefore helps run a quick evaluation (12-layer, 768-hidden, 12-heads, 110M parameters).

2. **M2 - Sarcasm**: The CASCADE model (Hazarika et al., 2018) is the current state of the art on the SARC dataset. This achieves a 77% balanced set accuracy and 79% unbalanced set accuracy. This model computes user-specific embeddings from their comments on other threads, thread embeddings from other user comments on the same thread, and the embeddings of the input text, and uses all of these for sarcasm detection. We use the CASCADE model but without the user and thread embeddings since they were not readily available for this dataset. The CASCADE model modified as above provides reasonable performance for the task of sarcasm detection.

3. **M3 - Humour**: We use an SVM model with bag of words features for humour detection as used in pre-existing implementations[2]. This provides an accuracy of 83% on the yelp reviews dataset.

4. **M4 - Hate Speech**: We use the implementation provided by (Davidson et al., 2017) which is simple Logistic regression model that provides a F1 score of 0.90 on **D4**.

M2, M3, M4 models as described above predict the probability of occurrence of sarcasm, humour and hate speech respectively on a given input text. These probabilities are then fed as features to our BERT-base sentiment classification model as described in **M1** above for the Yelp reviews dataset. We compare our modified model with BERT-base and observe a small improvement in the sentiment detection performance which positively supports our hypothesis **H1**. This motivates us to consider

[2] https://github.com/srishti-1795/Humour-Detection

developing embeddings for each of these language specific features instead of using just the probability of their occurrence.

3.3 Validating hypothesis H2

In order to test hypothesis **H2**, we construct a general-purpose feature embedding model E for all the four tasks, along with a classifier C for classification on the combined representation, as discussed below.

3.3.1 Feature Embedding Model

Given a d-word sentence s, we initialize trainable 128-dimensional word embeddings for each word, and create a $d \times 128$ embedding matrix for the sentence. In order to capture n-gram features, we use a word-level Convolutional Neural Network (Kim, 2014) with f filters $(n \times 128)$ for $n = 3, 4, 5$. For each n, we compute the $(d-n+1) \times f$ output, and use max pooling to get a f-dimensional vector. We concatenate these vectors for all three values of n and flatten them to get a $3f$-dimensional embedding. With $f = 100$, we get a 300-dimensional embedding for every sentence.

We use one such model for each of Sentiment (**E1**), Sarcasm (**E2**), Hate Speech (**E3**), and Humour (**E4**).

3.3.2 Combination Classifier

Given a d-word sentence s and a set of m feature embedding models $(1 \leq m \leq 4)$ $E \subseteq [\mathrm{E1}, \mathrm{E2}, \mathrm{E3}, \mathrm{E4}]$, we calculate a set of 300-dimensional embeddings per model, and concatenate them into a single $300 * m$-dimensional feature vector v. This is used as input to a sentiment classifier that predicts $\mathbf{C}(v) \in [0, 1]$ that represents the probability of positive sentiment. This classifier consists of two fully-connected layers with hidden size 50 and ReLU activation. This model is trained and tested separately for several combinations E, and the results from these experiments are discussed in section 4.

4 Experiments and Results

4.1 Results for Hypothesis H1

To test if our hypothesis **H1** holds true, we concatenated predictions from the sarcasm (M2) detection model to the BERT-base model embeddings (M1) used for sentiment classification as described in section 3.2. We used the PyTorch implementation of BERT Base[3], and our results are tabulated in Table 2. We trained the models for 3 epochs with a learning rate of 2e-5 and a batch size of 32.

Model	Accuracy (%)
M1	95.13
M1 + M2 (P/L)	**95.22**

Table 2: Testing our hypothesis: M_i refers to the respective model, P and L indicate using predicted probabilities and labels respectively. Both of our augmented models perform better than the sentiment model alone (M1), thus validating our hypothesis.

This experiment validated our hypothesis that there is tangible sentiment information to be gleaned from a sentence's sarcasm features (and potentially other features as well).

Attribute Combination	Accuracy (%)
Se	95.95
Se + Sc	96.02
Se + Hu	95.93
Se + Ha	95.69
Se + Sc + Hu	96.06
Se + Sc + Hu + Ha	**96.18**

Table 3: Model Performances for various combinations of Sentiment Se (E1), Sarcasm Sc (E2), Humor Hu (E3) and Hate Speech Ha (E4). We find that our combined model performs the best.

4.2 Results for Hypothesis H2

As described in 3.3.1, we used a single model architecture for training separate models on sentiment, sarcasm, humor and hate speech. Due to class imbalance and large dataset sizes, we modified our datasets in the following ways:

- **D1**: We took a subset of the training set for Yelp reviews which amounted to 100k reviews for training and validation combined, and included the entire test set of 36.6k reviews for reporting the performance of our models.

- **D2**: To maintain consistency, we took a sample of 100k comments for training our model on sarcasm detection.

[3] https://github.com/huggingface/pytorch-pretrained-BERT

Attribute Combination	**Accuracy (%)**		
	Run 1	**Run 2**	**Run 3**
Se	95.95	95.76	95.85
Se + Sc + Hu + Ha	**96.18**	**96.08**	**96.40**

Table 4: Model performances during several runs of the baseline and the combined (Se + Sc + Hu + Ha) models.

Review	**NLU Aspect Missed**
...The manager went around and asked the 2 waitresses working all 4/5 tables surrounding us and none of them took responsibility or seemed to want our table. Little did we know this was a blessing in disguise...	Extended story line: the first half of the review is negative, and the model likely misses the turning point towards positive halfway through the review

Table 5: Examples where the combined model goes wrong and the baseline sentiment model predicts the correct sentiment.

Review	**NLU Aspect Missed**
Say it with me now: Blaaaaaaaaaand	Indicative word missed
This place is closed, and for good reason.	Flip in sentiment

Table 6: Examples where the combined and the baseline sentiment models both fail to predict the correct sentiment.

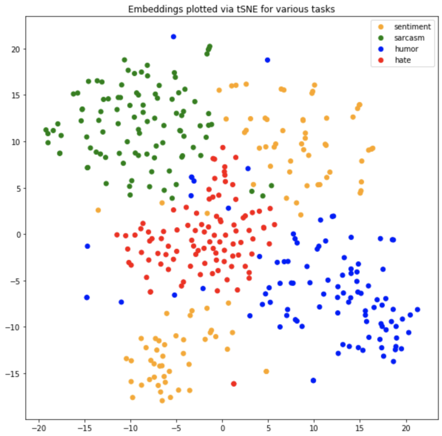


Figure 3: Scatter plot for the various task-specific embeddings (unit normalized) of the first 100 test reviews from the Yelp dataset. It can be seen that sentiment, sarcasm, humor and hate predominantly occupy different regions of the embedding space when reduced to 2D using tSNE.

- **D3**: Since this dataset had a ratio of 19:1 for non-humorous to humorous reviews, we took the entire set of humorous reviews from the original Yelp dataset, and added about twice the amount of randomly sampled non-humorous reviews to maintain a healthy ratio of 2:1. Final number of train/val reviews were 40k and test reviews were 8,661.

- **D4**: The number of hateful tweets in this dataset were 16 times the number of non-hateful tweets. We oversample the non-hateful tweets by 4x, and undersample the hateful tweets by 2x to obtain a ratio of 2:1 in favor of the positive class, for both the training/val and the test datasets. The total number of tweets in train/val are 16,854 and number of test tweets are 3,521.

After training a model each on these datasets following the architecture described in 3.3.1, we obtain the sentiment, sarcasm, humor and hate embeddings for each of the training and testing reviews from **D1**. For this model, we use CNN window sizes of 3, 4 and 5 with 100 kernels each, batch size of 64, learning rate 0.001, and dropout probability 0.5.

Further, we train combined sentiment classifiers (3.3.2) on top of various combinations of subsets of these embeddings for the training set, and then evaluate performance on the test set. The performances of these combinations is reported in Table 3, and results from more runs comparing the baseline against our combined model are shown in 4. In order to test if the improvement of our proposed hybrid model is consistently better than the baseline, we train our the combined and baseline models over fixed training set size multiple times and evaluate the performance on the same held-out test set consisting of 36,614 reviews. Table

4 shows the results of the models on the test set. We observe a consistent improvement using sentiment, sarcasm, humour and hate speech features as compared to just sentiment features.

5 Discussion and Analysis

Our hypothesis **H1** is supported by the experiments in both 4.1 and 4.2, i.e. sarcasm, humor and hate speech are signals that boost the sentiment classification performance on Yelp reviews.

5.1 Different natural language features add mutually exclusive information

As Figure 3 shows, the normalized task-specific embeddings occupy distinct regions of the 2D space (after dimensionality reduction using tSNE). Thus, the three additional models probably assist the sentiment embeddings by combining information from the source review that the sentiment model may not have learned to catch, and that might, in certain cases, help the combined model make better decisions. Since these embeddings have been obtained via a single model architecture trained on different datasets, the increase in performance on sentiment classification validates our hypothesis **H2**. This implies not only that a single architecture might suit multiple natural language problem domains, which models like BERT have already shown, but also that one or more of them can help boost the performance of others, if the reasoning behind such a predicted improvement is linguistically sound.

5.2 Interpretibility of model's success modes

We analyze the contribution of each of the individual models trained for sarcasm, humour and hate speech detection to the performance of sentiment detection by comparing the predicted labels against the ground truth. In each of the matrices in Figure 4, the value in cell (i, j) of category c (which can be TP, TN, FP or FN) denotes the respective fraction of predictions in model i that belong to category c if the predictions of model j are assumed to be the ground truth.

As evident from Figure 4, we find that adding features pertaining to sarcasm, hate speech, and humour to a baseline sentiment classifier increases the number of true positives against the Ground Truth labels (the baseline model predicts 97% of the combined model's positive labels). Since the false positive rate of the combined model (0.02) is also less than that of the sentiment model (0.03) against the ground truth, this shows that the *combined model has a higher precision*. This observation is also consistent with the reasoning that *sarcasm and hate speech are both likely to catch negative reviews which may otherwise sound positive to a naive model*, and thus reduce the false positive rate.

Similarly, looking at the true negatives' matrix shows that the combined (Sentiment + Sarcasm + Humour + Hate Speech) model predicts negative labels better than the (Sentiment + Sarcasm + Humour) model (0.97 against 0.95), which is consistent with the reasoning that *Hate Speech is likely to indicate a negative sentiment*, and that knowledge of Hate Speech helps the model better understand what to look for in a review that is negative because of hateful language.

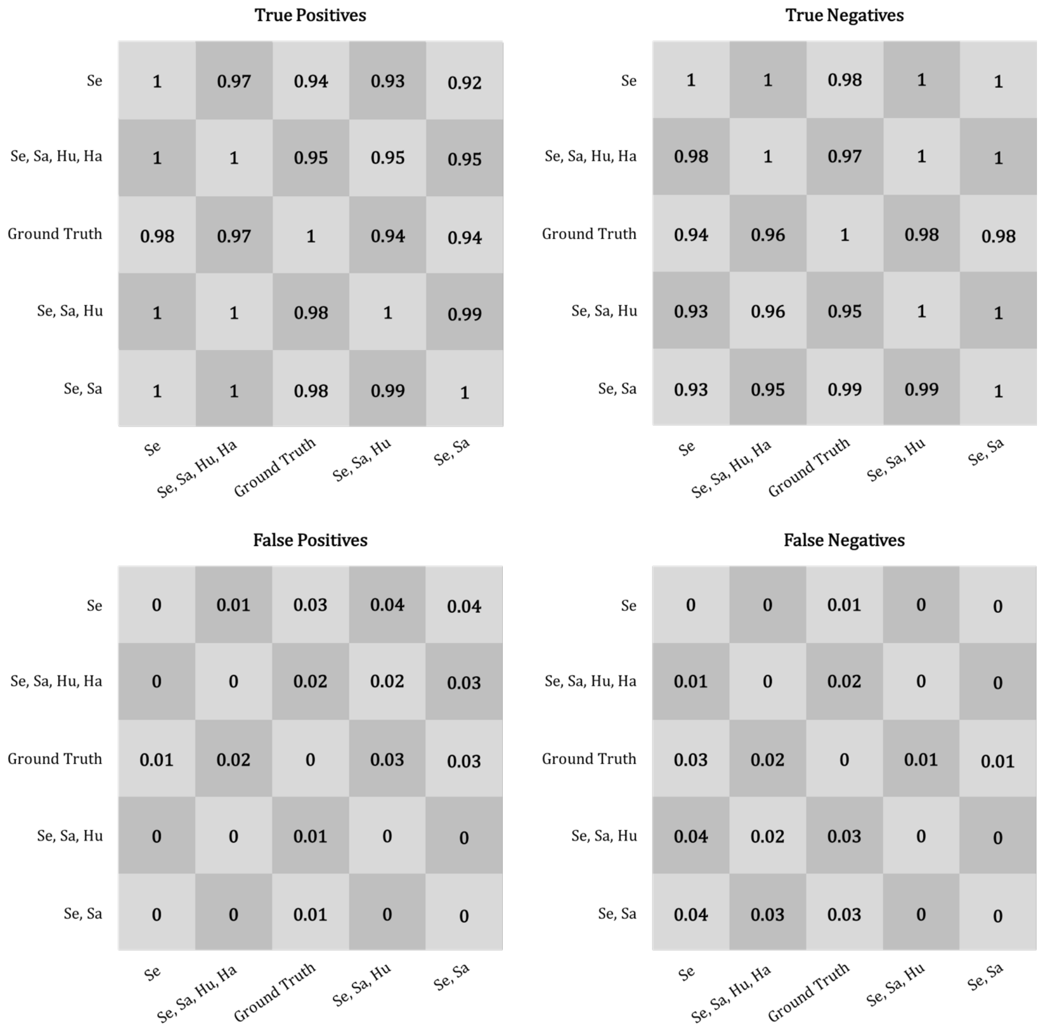
On looking at some reviews in the test dataset that our combined model (Se + Sc + Hu + Ha) got right and that the baseline sentiment model got wrong, we observe that our model definitively helps with identifying the right sentiment for sarcastic reviews (Table 1) and some hate reviews, although we couldn't find many humorous reviews in this context. Similarly, it seems that the reviews whose sentiment has been classified wrongly by the sentiment-only baseline and that don't have any sarcastic / hate intent don't get classified correctly by our combined model either (Table 6).

5.3 Error Analysis

Tables 5 and 6 present some analysis of the errors made by our model, and how they compare with the baseline Sentiment model. As can be seen, our model doesn't catch the elements of natural language that it was not trained to detect, and while it is quite sensitive to catching negative sentiment, it doesn't do as well when sentiment changes halfway through the review.

6 Conclusions

In this paper we show that features from sarcasm, humor and hate speech help in improving sentiment classification performance on the Yelp reviews dataset. We have also shown that a general-purpose model architecture for binary classification can be trained on each of these natural language tasks individually and that it provides an easy way for end-to-end sentiment classification that combines the strengths of each of these mod-



True Positives

	Se	Se, Sa, Hu, Ha	Ground Truth	Se, Sa, Hu	Se, Sa
Se	1	0.97	0.94	0.93	0.92
Se, Sa, Hu, Ha	1	1	0.95	0.95	0.95
Ground Truth	0.98	0.97	1	0.94	0.94
Se, Sa, Hu	1	1	0.98	1	0.99
Se, Sa	1	1	0.98	0.99	1

True Negatives

	Se	Se, Sa, Hu, Ha	Ground Truth	Se, Sa, Hu	Se, Sa
Se	1	1	0.98	1	1
Se, Sa, Hu, Ha	0.98	1	0.97	1	1
Ground Truth	0.94	0.96	1	0.98	0.98
Se, Sa, Hu	0.93	0.96	0.95	1	1
Se, Sa	0.93	0.95	0.99	0.99	1

False Positives

	Se	Se, Sa, Hu, Ha	Ground Truth	Se, Sa, Hu	Se, Sa
Se	0	0.01	0.03	0.04	0.04
Se, Sa, Hu, Ha	0	0	0.02	0.02	0.03
Ground Truth	0.01	0.02	0	0.03	0.03
Se, Sa, Hu	0	0	0.01	0	0
Se, Sa	0	0	0.01	0	0

False Negatives

	Se	Se, Sa, Hu, Ha	Ground Truth	Se, Sa, Hu	Se, Sa
Se	0	0	0.01	0	0
Se, Sa, Hu, Ha	0.01	0	0.02	0	0
Ground Truth	0.03	0.02	0	0.01	0.01
Se, Sa, Hu	0.04	0.02	0.03	0	0
Se, Sa	0.04	0.03	0.03	0	0

Figure 4: Overlap of model predictions with one another. Adding features from Sarcasm, Hate Speech, and Humour to baseline Sentiment classifier improves its ability to predict True Positives against the Ground Truth labels.

els.

This work shows that natural language understanding problems need not be thought of in isolation of each other. When motivated by human insights on how language is perceived, solutions to nuanced sub-problems might help solve more general problems like sentiment classification.

Future Directions

An interesting future direction is to test the combination of features from sarcasm, humor and hate speech for more fine-grained sentiment classification, as in Yelp (5-way classification) or Stanford Sentiment Treebank (5-way classification). We believe that our formulation would help in this case, by distinguishing between 1-star and 2-star reviews based on offensive language for instance. It would also be interesting to see if the same generic template achieves state of the art performance on other classification tasks like entailment detection, entity detection, etc.

Acknowledgements

The authors would like to thank Prof Christopher Potts from Stanford NLP Group for valuable suggestions and discussion on experimental evaluation and analysis. The authors would like to thank Min Kim for his suggestions on the experimental setup and the paper writeup. The authors would also like to thank the authors of CNN sentence classification (Kim, 2014) for providing open-source implementations of their models which allowed us to bootstrap our research ideas.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. *CoRR*, abs/1706.00188.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ona de Gibert, Naiara Perez, Aitor García Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 581–586, Stroudsburg, PA, USA. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: Contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1837–1848, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm. *CoRR*, abs/1704.05579.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Y. Alex Kolchinski and Christopher Potts. 2018. Representing social media users for sarcasm detection. *CoRR*, abs/1808.08470.

Mika Viking Mäntylä, Daniel Graziotin, and Miikka Kuutila. 2016. The evolution of sentiment analysis - A review of research topics, venues, and top cited papers. *CoRR*, abs/1612.01556.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. *CoRR*, abs/1610.08815.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Robert West, Hristo S. Paskov, Jure Leskovec, and Christopher Potts. 2014. Exploiting social network structure for person-to-person sentiment analysis. *CoRR*, abs/1409.2450.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.