What Matters in Irony Detection: An Extended Feature Engineering for Irony Detection in English Tweets.

Linrui Zhang, Qixiang Pang, Belinda Copus
University of Central Missouri
W.C. Morris 222, Warrensburg, Missouri
{zhang, qpang, copus}@ucmo.edu

Abstract

In recent years, large-scale language models (LLMs) have nearly become the dominant force in almost every natural language processing (NLP) task. The primary research approach has focused on selecting the most appropriate language model for specific NLP tasks and then incorporating linguistic features to enhance the model’s performance. With swift progress in this field, new features and models are evolving rapidly, and outdated systems require timely updates. In this paper, we extended the accomplishments of SemEval-2018 Task 3, enhancing its irony detection systems with novel features and more sophisticated language models. Subsequently, we conducted an ablation study to showcase the contributions of these enhancements to the LLM-based system. Furthermore, we compared our leading system with the top performers in the SemEval-2018 competition, and our best model exhibited superior performance compared to the leading performers applied to the same corpus.

Introduction

The identification of irony has a lengthy history involving the utilization of linguistic features, whether in the context of traditional rule-based or machine learning-based approaches (Joshi, Bhattacharyya, and Carman 2017). These features often include lexical features, sentiment features (Bouazizi and Ohtsuki 2015) (Farías, Patti, and Rosso 2016), (González-Ibáñez, Muresan, and Wacholder 2011), Part-of-Speech (POS) tags (Reyes, Rosso, and Veale 2013) and so on. In the past few years, with the advancement of Deep Learning, researchers have developed a standardized two-step pipeline for constructing irony detection models: (1) performing feature engineering and (2) inputting the extracted features into neural network architectures, such as RNN (Schmidhuber 1989), LSTM (Hochreiter and Schmidhuber 1997). For example, in SemEval-2018 Task 3 (Van Hee, Lefever, and Hoste 2018), the second-ranked system, THU_NGN (Wu et al. 2018), was constructed using a Dense-LSTM model that incorporated POS tags and sentiment features. However, in recent years, with the rise of the pre-training and fine-tuning paradigm, there has been a growing number of proposals for pre-trained large-scale language models like BERT (Vaswani et al. 2017), and they are increasingly assuming a prominent role in the field of Natural Language Processing. This shift has led to the phasing out of proposed models from SemEval-2018 Task 3. In this case, we have chosen novel features—those either not selected or only partially selected by SemEval teams—and applied them across a range of cutting-edge language models, including BERT, XLNET, and their variations. Subsequently, we demonstrated the impact of these newly incorporated features and language models on the task of irony detection, and compared our renovated models with the top performers in SemEval-2018. The primary contributions of this paper are as follows:

1. We enhanced the SemEval-2018 work by incorporating novel features and cutting-edge language models.
2. We demonstrated the impact of these enhancements on the LLM-based irony detection system.
3. Our best system outperformed the top participants in the SemEval-2018 Task 3 corpus.

Task Description

Irony detection refers to the process of identifying and understanding instances of irony in written or spoken language. The inception of this task traces back to SemEval-2018 Task 3 (Van Hee, Lefever, and Hoste 2018), which was the first shared task on irony detection. In this initiative, ironic tweets were gathered through the use of irony-related hashtags (i.e. #irony, #not) and were subsequently annotated manually. There are two goals for this task: (1) Task A involves determining whether a given tweet is ironic (Binary Classification), and (2) Task B involves identifying which type of irony (if any) is expressed (Multilabel Classification). For example, consider the following tweet:

A wonderful day of starting work at 6 am.

The phrase “wonderful day” typically implies a positive or enjoyable experience. However, the addition of “starting work at 6 am” suggests an early and potentially undesirable or challenging start to the day. Thus, this tweet is classified as ironic (Task A), and its type is verbal irony realized through a polarity contrast (Task B).
Feature Selected
We have chosen four linguistic features and five language models with the potential to enhance the performance of the irony detection model. The selected components and the rationale behind their selection are outlined as follows:

- **Emoji.** Emojis serve as visual indicators of emotions, tone, or sarcasm, helping to disambiguate the intended meaning of a message. By incorporating emojis into irony detection models, they can leverage this additional layer of information, making it easier to discern between literal and ironic expressions (Shiha and Ayvaz 2017) (Chen et al. 2018).

- **Emojitext.** (Singh, Blanco, and Jin 2019) has shown that replacing emojis with their natural language description can significantly improve accuracy for tweet classification. Applying this concept, we utilized the Emoji for Python package\(^1\) to transform emojis and emoticons into text representing their meanings. For example, 😁 will be converted into “smiling face”. This approach enables us to gather additional context from tweets.

- **Domain data.** The essence powering machine learning models lies in data (Li, Hou, and Che 2022). Numerous studies (Wei and Zou 2019) (Liu and Yu 2020) (Dong et al. 2021) have demonstrated that augmenting domain-specific data enhances the efficacy of the training process, resulting in improved model performance. As a result, we integrated additional data from the iSarcamEval corpus (Farha et al. 2022), specifically focusing on the sarcasm detection task sourced from SemEval-2022 Task 6. Given the resemblance between sarcasm and irony in text, we assumed that this incorporated corpus could boost the language model’s ability to generalize, alleviate overfitting, and thereby enhance the overall system performance.

- **Hashtag.** Hashtags serve as contextual cues on social media platforms. In the context of irony, hashtags such as #irony, #sarcasm, and #not provide users with valuable signals by highlighting the intended ironic tone or context in a concise and recognizable manner.

- **Language models.** We have selected five prominent language models: BERT (Vaswani et al. 2017), BERTweet (Nguyen, Vu, and Nguyen 2020), TwHIN-BERT (Zhang et al. 2022), ALBERT (Lan et al. 2019), and XLNet (Yang et al. 2019). Notably, BERT serves as our baseline model. BERTweet and TwHIN-BERT represent variants of BERT specifically trained with Tweet language, illustrating the impact of word embeddings on the system. ALBERT, a lighter version of BERT, is utilized to demonstrate the influence of parameter size on the system. Additionally, we have included XLNet, which is not part of the BERT family, to enrich the diversity of the chosen models.

System and Approach

Data Preprocessing
The initial SemEval irony detection corpus provided 3,834 English tweets for training and an additional 784 tweets for testing. To enrich the dataset, we integrated data from the iSarcamEval corpus (Farha et al. 2022), expanding the training set with 4,335 instances of (non-)sarcastic data. Emojis and hashtags were already managed by the original corpus—removed or added as necessary. For converting emojis (e.g., 😞) into their text descriptions (e.g., “angry face”), we utilized the Emoji for Python project. Additionally, we implemented a data cleaner program to rectify or eliminate incorrect, corrupted, improperly formatted, duplicate, or incomplete data within the dataset.

\(^1\)https://pypi.org/project/emoji/

Figure 1: The main structure of our system.

System Overview
We utilized five large-scale language models: BERT, BERTweet, TwHIN-BERT, ALBERT, and XLNet. Here, we use the BERT model as an example to illustrate the fundamental structure of our system. The pipeline of the structure can be found in Figure 1.

We initially employed the previously discussed method to process the tweets and then passed them to the tokenizer. Using the pre-trained BERT tokenizer, the input tweets underwent tokenization, breaking them into smaller tokens in preparation for input into the BERT encoder. Subsequently, the BERT encoder generated a condensed representation summarizing the entire input sequence. Finally, this representation was fed into a classification layer for the ultimate task of determining whether the given tweets are ironic or not.

We adopted the implementation of the BERT tokenizer and encoder from the Hugging Face Model Hub (Wolf et al. 2020) and initialized the tokenizer and encoder with the Bert-base-uncased checkpoint. We fine-tuned it for a maximum of 10 epochs, employing an early stopping
Figure 2: The performance of the irony detection models with the four chosen linguistic features and five language models on SemEval-2018 Task 3 Task A.

The Impact of Pre-trained Language Models

From Figure 2, we can observe that the top-performing models are BERTweet and TwHIN-BERT, with BERTweet particularly standing out as it clearly outperforms the other models. This distinction is due to the fact that the above-mentioned two models are pre-trained on a large corpus of tweets, which allows them to capture the specific linguistic nuances and characteristics of Twitter language. Regular BERT, on the other hand, is trained on a diverse range of text from the internet, which may not adequately capture the unique features of tweets.

In addition, we observed that ALBERT exhibits inferior performance compared to BERT and XLNet. This discrepancy can be attributed to ALBERT being a lightweight version of BERT\(^2\), featuring smaller models with fewer parameters and training on a less extensive dataset. The reduced capacity of ALBERT limits its ability to acquire richer and more detailed contextual representations compared to BERT and XLNET, leading to less than satisfactory performance. However, on the positive side, lightweight language models require less computational power and achieve faster training times.

The Impact of Emoji, Emojitext, Hashtag

Illustrated in the first three columns of Figure 2, emojis and emojitext can enhance system performance. However, their positive impact is negligible, and in some cases, it may even have a negative influence. Our understanding is that the influence of emojis (and emojitext) is absorbed by the BERT and XLNET encoder. Since they are not specifically trained on tweet text, it does not attribute meaningful embeddings to emojis. Instead, it treats them as normal words (or even noise), making it challenging to extract substantial information from them. Conversely, the outcomes appear promis-

\(^2\)The BERT-base model contains 110 million parameters, while ALBERT, with only 11 million parameters, is 10 times smaller than BERT.

Figure 3: The impact of domain-specific data on the irony detection systems for SemEval-2018 Task 3 Task A

Experimental Results and Analysis

Figure 2 illustrates the performance of the irony detection models with the four chosen linguistic features and five language models on Task A.
Teams
THU-NGN
0.536
0.723
UCDCC
NTU-SLP
0.672
0.705
0.724
BERTweet
0.496
TwHIN-BERT
NTUA-SLP
0.550
0.507
0.813
F1
0.495
WLV
UCDCC
F1
TwHIN-BERT
0.650
BERTweet

scenario.

rameters in their structures are not at the same level as the
size of their training set and the number of trainable pa-
models on external datasets. Nevertheless, both the
deemed employed the transfer learning approach, pre-training
round mid-2017. The leading participants in SemEval in-
date (Taigman et al. 2014) (Antol et al. 2015). The achieved
scores exceed those of the 1st ranked performer, UCDCC
(Ghosh and Veale 2018), in both tasks. This is primarily
attributed to our use of more powerful language models.
Even though the concept of transfer learning was estab-
established as early as 2014 (Taigman et al. 2014) (Antol et al.
2015), the widespread adoption of large-scale models began
around mid-2017. The leading participants in SemEval in-
deed employed the transfer learning approach, pre-training
their models on external datasets. Nevertheless, both the
size of their training set and the number of trainable pa-
parameters in their structures are not at the same level as the

It’s important to note that the iSarcamEval corpus lacks has-
tag information; thus, we have excluded it from the comparison
scenario.

Hashtags serve as significant indicators reflecting peo-
people’s attitudes, and users frequently use them to convey spe-
cific themes or topics, such as celebrating achievements with
#BossMoves or expressing indifference with #IDontCare. In
fact, in many studies on human attitudes, hashtags serve as
the gold standard for data collection (Rosenthal, Farra, and
Nakov 2019) (Ghosh et al. 2015) (Farha et al. 2022). There-
fore, it is unsurprising to observe, as depicted in the last three
columns of Figure 2, that the inclusion of hashtags signifi-
cantly increases system performance.

The Impact of Domain Data

Figure 3 illustrates the impact of domain specific data on the
system for Task A. The orange color indicates the perfor-
mance improvements attained through the utilization of data
from iSarcamEval corpus (Farha et al. 2022). From the fig-
ure, we can observe that incorporating (non-)sarcastic data
benefits the system performance of each language model un-
der three scenarios (i.e., base, base+emoji, base+emojitext) 3.
Considering the similarity that both irony and sarcasm
tasks involve a gap between the literal meaning of the words
and the intended meaning, the iSarcamEval corpus qualifies
as domain-specific data and thus contributes to the perfor-
mance gains.

Comparison with the SemEval-2018 Participants

We also compared our best system with the top perfor-
ners in SemEval-2018 Task 3 (Ghosh and Veale 2018) (Wu
The comparison results can be found in Table 1. It should
be emphasized that we did not leverage domain data to en-
hance our system for Task B since it involves a multiclass
irony classification problem, whereas the iSarcamEval cor-
pus focuses solely on binary classification. Furthermore, in
the original SemEval-2018 competition, hashtags were ex-
cluded from the training data. To maintain consistency in
our comparison, we also employed our non-hashtag models
for evaluation.

From the comparison results depicted in Table 1, we ob-
erved that our best model achieved F-1 scores of 0.813
and 0.550 in Task A and Task B, respectively. The achieved
scores exceed those of the 1st ranked performer, UCDCC
(Ghosh and Veale 2018), in both tasks. This is primarily
attributed to our use of more powerful language models.
Potential Improvement and Future Work

There are several potential improvements that we did not
pursue in the experiment due to constraints in time and hard-
ware. For instance, we opted not to utilize the more re-
cent language model GPT-3 (Brown et al. 2020), given its
substantial 175 billion parameters, which present a notable
challenge for Google Colab. In terms of linguistic features,
there are also several additional aspects to consider, such
as misspelled words or negation words like “a loooot of”
or “so goood”. As for domain-specific data, contemplating
the utilization of the Twitter sentiment analysis corpora from
past SemEval competitions (Ghosh et al. 2015) (Rosenthal,
Farra, and Nakov 2019) is also a possibility. These short-
comings will be deferred for future investigation.

Table 1: The comparison between our top performed irony
detection models with the leading teams in SemEval-2018 Task 3.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Teams</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BERTweet</td>
<td>0.813</td>
</tr>
<tr>
<td>2</td>
<td>UCDCC</td>
<td>0.724</td>
</tr>
<tr>
<td>3</td>
<td>TwHIN-BERT</td>
<td>0.723</td>
</tr>
<tr>
<td>4</td>
<td>THU-NGN</td>
<td>0.705</td>
</tr>
<tr>
<td>5</td>
<td>NTUA-SLP</td>
<td>0.672</td>
</tr>
<tr>
<td>6</td>
<td>WLV</td>
<td>0.650</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>Teams</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BERTweet</td>
<td>0.550</td>
</tr>
<tr>
<td>2</td>
<td>TwHIN-BERT</td>
<td>0.536</td>
</tr>
<tr>
<td>3</td>
<td>UCDCC</td>
<td>0.507</td>
</tr>
<tr>
<td>4</td>
<td>NTUA-SLP</td>
<td>0.496</td>
</tr>
<tr>
<td>5</td>
<td>THU-NGN</td>
<td>0.495</td>
</tr>
</tbody>
</table>

latest language models, such as BERT. For example, the
3rd performer, NTUA-SLP (Baziots et al. 2018), pretrained
their model on the SemEval2017Task4A dataset (Rosen-
thal, Farra, and Nakov 2019), which consists of only 50,333
tweets. This is notably smaller when contrasted with the to-
total training corpus of around 3.3 billion words used for mod-
models like BERT. Larger training datasets and parameters mean
greater model capacity or complexity, resulting in the supe-
ior performance of our model over the SemEval-2018 par-
ticipants.

Conclusion

In this paper, we have built upon the achievements of the
SemEval-2018 Task3 teams, enriching their studies by inte-
grating additional linguistic features and utilizing more re-
cent language models. We showcased the effects of these
modifications on the irony detection task. Through exper-
iments, we highlighted the performance improvements asso-
ciated with these features (and language models) and offered
explanations for the observed results. Additionally, we for-
mulated our own irony detection model, surpassing the per-
formance of the leading systems in SemEval-2018.
References


Bouazizi, M., and Ohitsu, T. 2015. Sarcasm detection in twitter:” all your products are incredibly amazing!!!”-are they really? In 2015 IEEE global communications conference (GLOBECOM), 1–6. IEEE.


