Zhang, S.; Zhang, X.; Chan, J.; Rosso, P. (2019). Irony detection via sentiment-based transfer learning. Information Processing & Management. 56(5):1633-1644. https://doi.org/10.1016/j.ipm.2019.04.006

Irony Detection via Sentiment-based Transfer Learning

Shiwei Zhang[a], Xiuzhen Zhang[a,*], Jeffrey Chan[a], Paolo Rosso[b]

[a] *RMIT University, Australia*
[b] *Universitat Politècnica de València, Spain*

Abstract

Irony as a literary technique is widely used in online text such as Twitter posts. Accurate irony detection is crucial for tasks such as effective sentiment analysis. A text's ironic intent is defined by its context incongruity. For example in the phrase "I love being ignored", irony is defined by the incongruity between the positive word "love" and the negative context of "being ignored". Existing studies mostly formulate irony detection as a standard supervised learning text categorization task, relying on explicit expressions for detecting context incongruity. In this paper we formulate irony detection instead as a transfer learning task where supervised learning on irony labeled text is enriched with knowledge transferred from external sentiment analysis resources. Importantly, we focus on identifying the hidden, implicit incongruity without relying on explicit incongruity expressions, as in "I like to think of myself as a broken down Justin Bieber - my philosophy professor." We propose three transfer learning-based approaches to using sentiment knowledge to improve the attention mechanism of recurrent neural models for capturing hidden patterns for incongruity. Our main findings are: 1) Using sentiment knowledge from external resources is a very effective approach to improving irony detection; 2) For detecting implicit incongruity, transferring deep sentiment features seems to be the most effective way. Experiments show that our proposed models outperform state-of-the-art neural models for irony detection.

Keywords: `Irony Detection, Transfer Learning`

*Corresponding author.
Email addresses: `shiwei.zhang@rmit.edu.au` (Shiwei Zhang), `xiuzhen.zhang@rmit.edu.au` (Xiuzhen Zhang), `jeffrey.chan@rmit.edu.au` (Jeffrey Chan), `prosso@dsic.upv.es` (Paolo Rosso)

1. Introduction

User-generated texts on social media platforms like Twitter and Facebook often involve the widespread use of creative and figurative languages like irony and sarcasm. A text utterance is perceived to be ironic if its intended meaning is opposite to what it literally expresses. The terms irony and sarcasm are often used interchangeably, despite their subtle differences in meaning [1]. Accurate irony detection is important for social media analysis. For example, failing to detect irony can lead to low performance for sentiment analysis, since the presence of irony often causes polarity reversal [2]. Also, irony detection is important for security services to discriminate potential threats from just ironic comments [3]. In contrast to most text classification tasks, irony detection is a challenging task [4] that requires inferring the hidden, ironic intent, which can not be achieved by literal syntactic or semantic analysis of the textual contents. Indeed the challenge of irony detection is clearly shown in the sentiment polarity classification task of Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (Evalia 2016) [5]. Three independent subtasks are included, namely subjective classification, polarity classification and irony detection. The performance for both subjective classification and polarity classification is over 10% higher than that of irony detection in terms of F Measure.

According to linguistics research, irony is the incongruity expressed between the context and statement conveyed in a piece of text [6, 7, 8]. Sentiment polarity contrast is a commonly seen form of irony on Twitter [9, 8]. For example in the tweet "I love when I wake up grumpy", "love" expresses positive polarity whereas the phrase "wake up grumpy" expresses negative polarity. The hidden sentiment polarity contrast signifies the ironic intent of the tweet. Moreover, the extent of irony perception depends on the strength of the context ("wake up grumpy") and the strength of the statement ("love"). Explicit incongruity refers to contrast from explicit sentiment words as in "I love being ignored", where "love" is positive and "ignore" is negative. Implicit incongruity refers to contrast from phrases expressing implicit sentiment polarity but not using explicit sentiment words, as in "I love this paper so much that I put in it my drawer"; the phrase "put in my drawer" implies a negative polarity and forms a contrast with the positive

sentiment in “love”.

The task of irony detection is to classify a piece of text as ironic or non-ironic. Existing studies mostly formulate irony detection as a standard supervised learning text categorization problem. Approaches to irony detection on Twitter can be roughly classified into three classes, namely rule-based approaches, classical feature-based machine learning methods and deep neural network models. In the literature, rule-based and classical feature-based machine learning models are proposed for irony detection (See [10] and [11] for surveys). Recently deep learning models are applied for irony detection [12, 13, 14, 15, 16, 17] and show better performance than classical feature-based machine learning models. Among all of the neural network-based models, attention-based models are most effective. Apart from standard attention models, a recent work [18] proposed an intra-attention mechanism for sarcasm detection. Their model is looking into intricate similarities between each word pair.

Most previous studies do not study context incongruity for irony detection. A few studies focus on identifying context incongruity for irony detection, but with limitations. One previous study [9] made use of the pattern of “positive sentiment followed by negative situation” to detect irony on Twitter. The approach can miss many forms of context incongruity that do not follow this pattern. Another previous study [8] manually engineered explicit and implicit context incongruity features for irony detection and still can capture limited context incongruity.

In this paper we formulate irony detection as a transfer learning task where supervised learning on irony labels is enriched with knowledge transferred from external sentiment analysis resources. Moreover, we focus on the key issue for irony detection – identifying the hidden, implicit incongruity without explicit incongruity expressions as well as the explicit incongruity. Our key idea is to transfer external sentiment knowledge from sentiment resources to train the deep neural model for irony detection. Resources for sentiment analysis are readily available, including sentiment lexica [19] and sentiment corpora [20]. We propose three sentiment-based transfer learning models to improve the attentional recurrent neural model for identifying explicit and implicit context incongruity for irony detection on Twitter. The three models are designed to transfer different types of sentiment knowledge. The first two methods are focused

on transferring hard sentiment attention generated from a pre-defined sentiment corpus, but the hard attention in the first model is treated as an external feature while it is treated as an extra supervised signal in the second model. The last model is focused on transferring deep features from the sentiment analysis on Twitter for the irony detection task, where features from both tasks are mapped into a common latent feature space. By comparing these different approaches one can find the most effective way of using sentiment-based transfer learning for irony detection.

Main contributions of this paper are:

- We find that leveraging sentiment knowledge from rich sentiment resources is an effective way to improving irony detection.
- Learning deep features on sentiment tweets corpora and transferring them into the attention-based neural model is the most effective way to detect both explicit and implicit context incongruity.
- To our best knowledge, for the first time, we contrast the human-labeled and hashtag-labeled datasets for evaluation of irony detection models. We find that the human-labeled dataset is much more challenging than the hashtag-labeled dataset and gives a more accurate estimation of the performance for irony detection models in real applications.

The rest of this paper is organized as follows: In Section 2, we state our research objective. In Section 3, we discuss the related work including both conventional methods and deep learning methods on irony detection. In Section 4, we discuss the shortcomings of using attention-based Bi-LSTM on irony detection. In Section 5, we present the details of our proposed approaches. In Section 6, we describe the experimental setup and discuss experimental results, and interpret results with attention-based visualization. In Section 7, we conclude our work.

2. Research Objective

Identifying context incongruity is the key to detect the ironic intent of Twitter posts. But in the literature, automatic sarcasm/irony detection is commonly formulated as a

supervised learning classification task. Given a collection of tweets annotated as either *ironic* or *non-ironic*, a classification model is trained on the annotated tweets' collection and is then applied to predict the label of unseen tweets. For instance, most previous works do not extract patterns of the context incongruity or provide a clear reasoning on detecting the context incongruity, especially when using deep learning models.

Although the irony intent is mainly expressed by incongruous sentiment between the context and the statement, the limited annotated resource is a barrier for a model to fully detect those sentiment patterns given the extremely various sentiment patterns available in human languages. On the other hand, sentiment resources are widely and readily available, which could be leveraged for irony detection. We formulate irony detection as a transfer learning task where supervised learning on irony labels is enriched with knowledge transferred from external sentiment analysis resources. Specifically, our research objective is to address the following two research questions:

- How to transfer different types of sentiment knowledge for irony detection?
- How to effectively use the transferred knowledge to detect the context incongruity, especially the implicit context incongruity?

3. Related Work

Approaches to irony detection on Twitter can be roughly classified into three classes, namely rule-based approaches, classical feature-based machine learning methods and deep neural network models. Rule-based approaches generally rely on linguistic features such as sentiment lexicon or hashtags to detect irony on Twitter [21, 4, 22]. Twitter uses hashtags to invert the literal sentiment in tweets [21]. The most popular hashtags for indicating irony include *#irony*, *#sarcasm* and *#not* [22]. The use of hashtags like "#sarcasm", is believed to be a replacement of linguistic markers such as exclamations and intensifiers.[23] Classical feature-based machine learning approaches use hand-crafted features [1] for irony detection, such as sentiment lexicon, subjectivity lexicon, emotional category features, emotional dimension features or structural features.

In recent years, deep learning-based approaches have been applied to irony detection, where (deep) features are automatically derived from texts using neural network models. Using the similarity score between word embeddings as features has shown an improvement for irony detection [13]. A convolutional neural network (CNN) was proposed in [12] for irony detection, which uses a pre-trained convolutional neural network for extracting sentiment, emotion and personality features for irony detection. There are also several studies that use CNN-LSTM structures [24, 17] for sarcasm detection. Another interesting work focuses on detecting rhetorical questions and sarcasm using CNN-LSTM also, but with an additional fully connected layer used for the purpose of taking Linguistic Inquiry and Word Count (LIWC) features [15]. These existing studies use the convolutional network to automatically derive deep features from texts for irony detection. Results of these deep learning approaches are generally better than classical feature engineering-based approaches.

Recently attention-based recurrent neural networks (RNNs) were proposed for irony detection [16, 25, 14] and other NLP tasks [26, 27, 28]. The self-attention mechanism is not directly targeted to identify context incongruity. One of the previous work [25] studied emotion, sentiment and sarcasm prediction, where the attention mechanism is not particularly used to detect context incongruity. Another previous work [16] studied irony detection for replies in social media conversions. The sentence-level attention mechanism is used to identify more informative sentences in conversations that trigger sarcasm replies. In addition, a previous work [14] focused on irony detection in tweets and employed the standard attention mechanism. However, the standard self-attention mechanism often generates attentions for only partial texts forming the context-statement contrast and thus fail to detect the context incongruity (More details in Section 3). A recent work proposed a neural network with intra-attention for sarcasm detection on social media, which is focusing on intricate similarities between each word pair in sentence [18]. Almost all of the previous work are using a handful human-labeled ironic tweets for training, however, pattern recognition for detecting irony is so complex and difficult that a considerable size of the dataset is needed. As it is costly to build a large annotated dataset for training a high-performance model, transfer learning with sufficient sentiment resources seems to be at hand as an alterna-

tive.

Irony detection via identifying context incongruity has been reported in the literature, but the proposed solutions very much rely on manually engineered patterns and features. In one of the previous work [9], sarcasm is identified via a pattern of "positive sentiment followed by negative situation", and a bootstrapping algorithm (originated from the word "love") to automatically learn phrases corresponding to the positive sentiment and negative situation respectively. In [8], four types of manually engineered features including lexical features, pragmatic features, implicit incongruity features and explicit incongruity features are used to train a model for irony detection. It must be noted that although irony detection needs to detect sentiment incongruity, it is different from detecting sentiment shift [29], where words and phrases change the sentiment orientation of texts as in "I don't like this movie".

Existing studies [12] that use sentiment analysis resources for irony detection lacks a principled approach of transferring the sentiment analysis knowledge. In [12], a comprehensive set of features, including sentiment, emotion and personality features, are extracted from sentiment analysis resources for irony detection. The model combines all features before the prediction layer in the neural network, which makes it unclear whether the sentiment features benefit detecting context incongruity or irony detection. In contrast, our work does not only use sentiment knowledge but also consider the reasoning how and why we incorporate them in a neural network. Specifically, we devoted to using sentiment knowledge and resources with reasoning and visualization-focused interpretation to show how our models detecting context incongruity.

4. Attention-based Bi-LSTM (Bi-LSTM)

In this section, we introduce attention-based Bi-LSTM for the sake of understanding our proposed models. Recurrent neural networks (RNNs) are designed to process sequences. The Long Short-Term Memory (**LSTM**) is a commonly used RNN unit proposed by [30] to overcome the gradient vanishing problem. In terms of the network architecture, the Bidirectional LSTM [27] is widely used, which has two layers of LSTM reading sequences forward and backward respectively. The output of Bi-LSTM

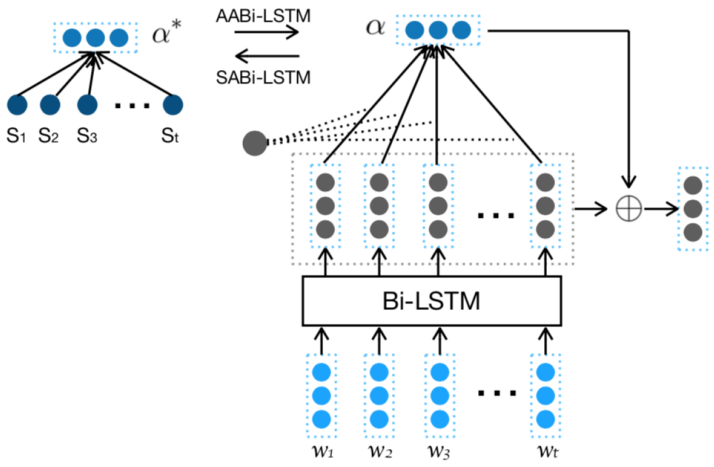

Figure 1: Sentiment Attention Bi-LSTM models. AABi-LSTM: model combines the hard sentiment attention with the learned soft attention. SABi-LSTM: model treats the hard sentiment attention as a supervised signal.

is a concatenation of forward and backward returned sequences:

$$h_i = [\overrightarrow{h}_i \parallel \overleftarrow{h}_i] \tag{1}$$

In the attention-based Bi-LSTM, H=$[h_1, h_2, ..., h_i]$ is a matrix consisting of output vectors produced by the Bi-LSTM, where i is the time step. The representation r of a tweet is formed by a weighted sum of these output vectors:

$$M = \tanh(H) \tag{2}$$

$$\alpha = softmax(\omega^T M) \tag{3}$$

$$r = H\alpha^T \tag{4}$$

$$h^* = \tanh(r) \tag{5}$$

At prediction, we use $softmax$ to predict $\hat{y}$ for a tweet. The goal of training is to minimize the cross-entropy error between the true label y_i and the predicted label $\hat{y}_i$:

$$\hat{y} = softmax(Wh^* + b) \tag{6}$$

$$loss^1 = -\sum_i \sum_j y_i^j \log \hat{y}_i^j \tag{7}$$

In equation (3), the vector α is the attention vector.

Bi-LSTM often fails to capture words and phrases crucial for building the ironic intent. This may be possibly due to the inherent difficulty of the task and limited annotated training instances.

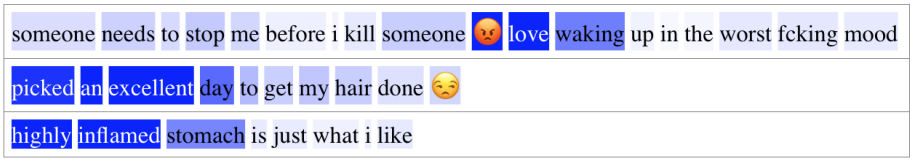

Figure 2: Examples of attention generated by the standard attention-based Bi-LSTM; the luminance of blue represents the attention value of each word.

As shown in Fig. 2, with the first tweet "someone needs stop me before kill someone ☹ love waking up in the worst fcking mood", Bi-LSTM only put strong attention on "loving waking" that indicates positive sentiment, and as a result failed to detect the negative sentiment expressed by "the worst fcking mood".

The failure of the standard attention mechanism for detecting context incongruity is possibly caused by the inherent difficulty of the task and only relying on the irony labels, which are limited. Moreover, LSTM even with attention can only learn the long dependences of the context. To detect context incongruity, we need to use external sentiment resources. In the next section, we describe our approach of enhancing the attention mechanism with sentiment knowledge transferred from the readily available resources for sentiment analysis.

5. Sentiment-based Transfer Learning for Irony Detection

Transfer learning is an important machine learning technique that takes advantages of the knowledge from solving one problem to solve other related problems, which can overcome the burden of limited human-labeled resources. Learning deep features or abstract representation of input is the advantage of deep learning used with transfer learning [31]. Transfer learning based models are particularly useful for cross-domain tasks. Especially when a target domain has very limited data, there is a need to train

a high-performance model using data in a source domain where data can be easily obtained [32]. Feature transformation can be completed by re-weighting a layer in the source domain to more closely match the target domain [33], or by mapping features from both source domain and target domain into a common latent feature space [34].

In our scenario, since detecting irony on Twitter is based on incongruous sentiment between the statement and the context, knowledge learned from the resources used for sentiment analysis will be incorporated into detecting irony. Sentiment analysis resources are widely available, including sentiment word corpora[35, 36] and sentiment tweets corpora[37, 38]. In order to improve the attention mechanism on detecting context incongruity, we propose our methods that are transferring sentiment knowledge from external resources, such as sentiment words corpora and sentiment Twitter corpus [19, 35, 36], as additional resources to enrich the limited human annotated ironic tweets. The challenge is how to represent and incorporate the sentiment knowledge into the attention mechanism for irony detection.

In order to incorporate two different types of sentiment resources into irony detection, namely sentiment word lexica and sentiment tweets copra, we propose different models to transfer different sentiment knowledge. The first two models are incorporating sentiment word lexica, where the sentiment-based hard attention is generated to strengthen the attention distribution on sentiment parts, but with different methods. The major difference between them is how the sentiment-based hard attention being incorporated. In the first model, the sentiment-based hard attention is treated as a feature while it is treated as a supervised signal in the second model. Being different from our first two models, our third model is proposed to detect context incongruity using the transferred deep features from the model learned on sentiment Twitter corpus instead of using the sentiment-hard attention.

5.1. Sentiment-Augmented Attention Bi-LSTM (AABi-LSTM)

With the first model, the readily available sentiment word corpora [19, 35, 36] is used as additional resources to generate a sentiment distribution, which then will be treated as an hard attention and transferred into the soft attention mechanism in order to push the attention-based model to focus on context incongruity. In particular, our model

incorporates not only the polarity but sentiment strength into the attention mechanism to capture the strength of incongruity in tweets, based on the linguistic principle of "the extent of irony perception depends on the strength of context and statement" [6, 7].

In our model AABi-LSTM as shown in Fig. 1, we first construct a sentiment hard attention based on the sentiment of each word. The sentiment scores of each word are generated by using pre-defined sentiment corpora. For a given tweet, the sentiment distribution $[\alpha_1^*, \alpha_2^*, \alpha_3^*, ..., \alpha_i^*]$ is generated by applying *softmax* on absolute sentiment scores of each word $[S_1, S_2, S_3, ..., S_i]$. Then, we transfer this sentiment hard attention into the attention-based model to enhance the attention to the sentiment part of a tweet. In our first proposed mechanism, the sentiment attention vector is added to the learned attention vector in the network, which results in directly strengthening the attention of the network on the sentiment part:

$$\alpha^* = softmax(|S|) \tag{8}$$

$$r = H(\alpha \oplus \alpha^*)^T \tag{9}$$

5.2. Sentiment-Supervised Attention Bi-LSTM (SABi-LSTM)

In order to detect the complete contextual incongruity, we further propose to take advantage of the widely available sentiment Twitter corpora [38, 37] to improve the attention mechanism in a supervised manner so as to capture the complete context for incongruity. With our second model sentiment-supervised attention Bi-LSTM (SABi-LSTM), we learn the abstract representation of polarity embedded in expressions without sentiment words and transfer these learned features into irony detection model for learning context incongruity.

As shown in Fig. 1, SABi-LSTM includes a sentiment attention mechanism that uses the sentiment hard attention in a supervised manner, which provides an alternative supervised signal to let the model learn features and attentions with reinforced attentions on sentiment parts. Technically, the attention value is used as an output of the model apart from class prediction, which will be then used in supervised training with sentiment hard attention as the true label. In order to let the network's attention be close to sentiment distribution, another loss function is defined to minimize the *cosine*

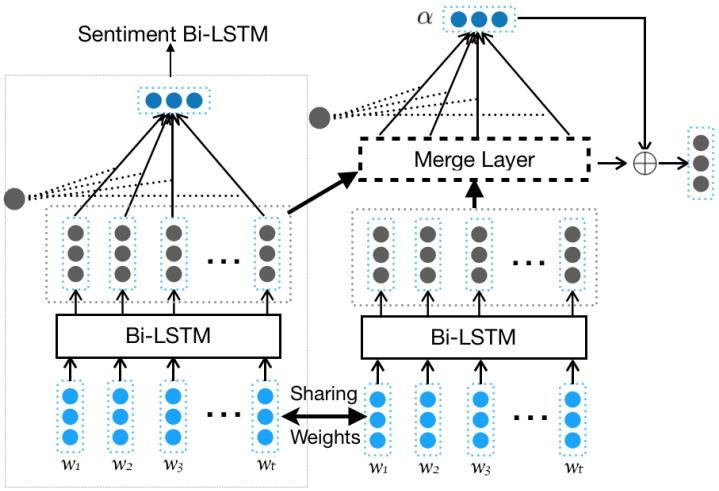

Figure 3: Sentiment transferred model (STBi-LSTM): has two training steps. 1. The sentiment Bi-LSTM is firstly trained on a sentiment corpus. 2. Two Bi-LSTMs are trained together on an irony corpus, but with the weight of sentiment Bi-LSTM frozen.

distance or $(1 - Cosine\_Similarity)$ between attention distribution and sentiment distribution, as follows:

$$loss^2 = 1 - \frac{\sum_{i=1}^{T} \alpha_i \alpha_i^*}{\sqrt{\sum_{i=1}^{T} \alpha_i^2} \sqrt{\sum_{i=1}^{T} (\alpha^*)_i^2}} \tag{10}$$

$$loss = loss^1 + \lambda * loss^2 \tag{11}$$

λ is a hyper-parameter to adjust $loss^2$ when updating neural network.

5.3. Sentiment Transferred Bi-LSTM (STBi-LSTM)

The previous two proposed models are transferring sentiment hard attention. Our third proposed method illustrated in Fig. 3 is designed to transfer deep features from sentiment analysis into irony detection for learning both explicit and implicit context incongruity. Our model consists of two Bi-LSTMs. one of Bi-LSTM acts as the sentiment feature extractor, while another one is the irony detector. The training process contains two parts. Firstly, the sentiment Bi-LSTM is trained on a readily available Twitter sentiment corpus, and then the weights of Bi-LSTM are kept frozen. In the

Table 1: Datasets

	Ironic vs. Non-ironic	Annotation
Reyes13	10,000 vs. 10,000	#irony vs. #education, #humor, or #politics
Barbier14	10,000 vs. 10,000	#irony, #sarcasm vs. #eduation, #humor, etc.
Ptacek2014	48890 vs. 18889	#sarcasm for sarcastic tweets
Riloff2013	1600 vs. 1600	manual
Moh2015	532 vs. 1397	manual
SemEval2018	2,396 vs. 2,396	manual

second part of the training, a tweet will be given to both Bi-LSTMs. The sentiment Bi-LSTM (or the sentiment feature extractor) outputs deep features that are about words with the implicit and explicit sentiment, and the second model firstly learns semantic features for the context. Both features are then mapped into a common latent feature space at Merger layer, and features on incongruous context are learned by the attention layer and the fully connected layer of the second Bi-LSTM. In terms of the mathematical operation of incorporating transferred deep features at Merger layer, it concatenates deep features from sentiment Bi-LSTM and the second Bi-LSTM before the attention mechanism:

$$H_{merged} = [H_{semantic} \parallel H_{sentiment}] \tag{12}$$

6. Experiments

We next discuss the experiment setup, including baselines and datasets, and then report results. We also report on results of error analysis for our models.

6.1. Baselines and datasets

We compared our models against deep learning-based irony detection models as well as representative conventional feature-based models.

- Bi-LSTM: Attention-based Bi-LSTM structure has been employed for irony detection in conversations and for learning representations for irony detection in

the literature [25, 16]. We implemented the network for our task and our implementation is based on the popular structure in [27].

- CNN-LSTM: Our implementation closely followed the architecture in [24]. It has three different neural network layers, a convolutional layer followed by 2 LSTM layers and a fully connected layer with the same hyper-parameter settings.

- LSTM: The model proposed in [14] is an attention-based LSTM for irony detection on Twitter.

- CNN: The Convolutional network [39] is widely used for classification problems.

- [9], [8] and [1] are classical feature-based irony detection models. Especially [9] and [8] are representative models focused on context incongruity.

Several datasets are widely used in the irony detection literature. There are two approaches to annotate sarcasm. Some datasets are automatically annotated by using sarcasm hashtags #irony, #sarcasm and #not. Other datasets are manually annotated by humans.

- Reyes2013 [40], Barbieri2014 [41] and Ptacek2014 [42] are datasets automatically annotated by hashtags. The sarcastic tweets and non-sarcasm classes are annotated by hashtags as shown in Table 1. Each pair of sarcasm and non-sarcasm class of tweets form a dataset for evaluating irony detection.

- Riloff2013 [9], Moh2015 [43] and SemEval2018 [44] are manually annotated Twitter datasets. SemEval2018 is the official dataset used for SemEval 2018 Task 3 (Irony detection in English tweets). Statistics of the datasets are shown in Table 1.

For each dataset, we randomly split it into 80% for training and 20% for testing, except SemEval2018 using official training and testing splits. The parameters are tuned on 10% random portion of the training data. For a fair comparison, following the

literature, macro average F_1 was used as the evaluation metric, except SemEval2018 where the binary F_1 was adopted.

For data preprocessing, we chose a customized Twitter tokenizer from the Natural Language Toolkit (NLTK) [1]. The word embeddings for all models have been initialized with pre-trained Glove [45] word vectors with 300 dimensions. The word-level sentiment scores are generated by NLTK with the help of a sentiment analysis tool VADER [36], which is designed for sentiment analysis of social media data, especially Twitter. Another great advantage of VADER is that it not only provides the polarity of words, but also gives the sentiment strength of words. Also, we adopted a sentiment emoji corpus [20]. The sentiment corpus for transfer learning used in our STBi-LSTM, is built based on two sentiment corpora used in SemEval 2017 Task 4 [37] and SemEval2015 Task 11 [38]. The hyper-parameters are selected using a grid search. The best dimensions of hidden states for all variants of Bi-LSTMs in our grid search is 200.

6.2. Evaluation

Table 2: Results (F_1) for Irony detection on hastag-annotated datasets

	Reyes2013			Barbier2014				Ptacek 2014
	edu	*hum*	*pol*	*edu*	*hum*	*pol*	*news*	
Bi-LSTM	94.01	95.54	96.32	94.12	94.86	98.62	96.24	83.12
CNN-LSTM	92.04	92.73	93.33	93.15	94.78	97.48	96.10	81.00
LSTM	92.30	89.50	89.00	94.03	94.23	97.56	96.11	82.86
CNN	93.35	93.44	94.66	94.12	95.53	98.31	96.28	81.23
AABi-LSTM	94.23	95.56	96.42	94.92	95.73	98.18	96.91	83.02
SABi-LSTM	94.65	**95.82**	96.15	94.21	95.16	98.34	96.41	84.00
STBi-LSTM	**94.69**	95.69	**96.55**	**94.95**	**96.14**	**98.62**	**96.92**	**84.20**
[1]*	90.00	90.00	92.00	90.00	92.00	94.00	96.00	82.00

*As reported in the relevant papers.

[1] https://www.nltk.org/

In order to have a clear idea about how the models perform on datasets that used different annotation strategies, we chose to report the experiment results in two separate tables. Table 2 reports the experiment results on datasets that were hashtag-annotated or has been labeled by using specific hashtags, such as #ionry or #sarcasm. Table 3 reports the experiment results on datasets that were annotated by crowdsourcing platforms or human.

According to Table 2, it is obvious that irony detection on hashtag-annotated datasets is not difficult since most models have achieved very promising results almost across every dataset involved in this work. Compared with the dataset in Reyes2013 and Barbier2014, the dataset Ptacek2014 seems more difficult, which have had a significant drop (around 10%) on performance for all of the models. The imbalanced classes of Ptacek2014 (18889/48890) is the major factor that highly affects the performance. Additionally, it is clear that neural models are much better than traditional machine learning models, such as feature engineering with SVM [1].

Table 3: Results (F_1) for Irony detection on manually annotated datasets

	Riloff2013	Moh2015	SemEval2018
Bi-LSTM	73.57	58.31	64.15
CNN-LSTM	70.56	59.22	61.16
LSTM	72.17	57.16	63.66
CNN	74.75	57.71	62.03
AABi-LSTM	75.39	61.54	64.28
SABi-LSTM	74.63	59.37	65.33
STBi-LSTM	**77.85**	63.70	**67.55**
[9]*	51.00	-	-
[8]*	61.00	-	-
[1]*	73.00	**66.00**	-
SemEval2018 Top 3	-	-	**70.54** / 67.19 / 65.00

*As reported in the relevant papers.

Table 3 presents experiment results on human annotated datasets. In general, attention-

based models work better than other models including other neural models and conventional machine learning methods.

On dataset Riloff2013, the proposed three models all achieved better results than the baselines, especially our third proposed model achieved the best result which has about 4% improvement over the Bi-LSTM.

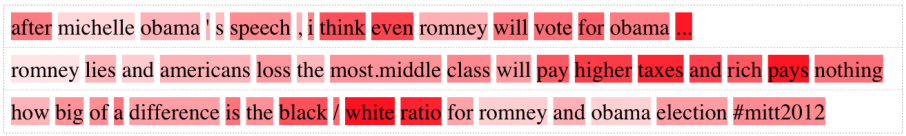

Figure 4: Examples of attention distribution on ironic tweets from Moh2015. Tweets in red are incorrectly classified. Attention are generated by our model STBi-LSTM.

Figure 5: Differences of attention distribution among attention-based models

In our experiments, results on dataset Moh2015 are the worst, which is not only because of the relatively small size of the dataset (1397 non-ironic tweets and 532 ironic tweets), but also because of the type of irony expressed in this dataset. This dataset was collected using hashtags related to the "2012 US presidential election", so that most of the ironic tweets are either situational irony or ironic utterance related to named entity or external knowledge.

For example, in Fig. 4, in order to detect irony expressed by those tweets, a model has to have the knowledge of named entity, such as "Obama" and "Romney", or back-

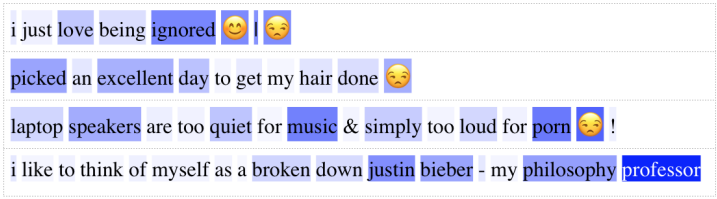

Figure 6: Examples of attention distribution learned by STBi-LSTM

ground knowledge of the tasked event "2012 US presidential election". However, our model (STBi-LSTM) is still able to capture most of the words or phrases with sentiment meaning. The dataset used by SemEval 2018 Task 3 "irony detection in English tweets" is another difficult task. Our proposed models have achieved better results than other baseline neural models, and our results are close to the top 3 in the official rank [44]. The best F_1 score is achieved by our third proposed model STBi-LSTM and it is better than the result of the second team in the official rank.

6.3. Discussion

We first discuss how our models improve the attention mechanism for detecting contexts for sentiment contrast. We further discuss how our third proposed model (STBi-LSTM) learns the contexts for explicit and implicit incongruity.

Fig. 5 presents an ironic tweet that has differently learned attention by our models and Bi-LSTM. Generally, Bi-LSTM is able to detect some of the words with sentiment meaning, but it seems often fail to detect the full context for incongruity. For example, in the first example of Fig. 5, it spreads very high attention on phrase "love waking up" expressing the positive sentiment, but with very tiny attention on phrase "worst fcking mood" which is the negative situation of this ironic tweet. In contrast, our proposed models all have more attention on "worst fcking mood". Most importantly, without using an explicit lexicon, our third proposed model STBi-LSTM is still able to detect both parts of context incongruity and spread balanced attention on both.

With the transferred deep features from the sentiment model, the STBi-LSTM performs very well, especially on detecting sentiment based context incongruity. We selected several examples of attention learned by STBi-LSTM in Fig.6. In this figure, the

first two tweets are examples of explicit context incongruity. “love” versus “ignored” is the key sentiment contrast in the first example, while “excellent day” versus “☹” is the sentiment contrast in the second example. In order to show the ability of our models on detecting implicit context incongruity, we picked two example tweets from the dataset. In Fig.6, the third tweet is ironic about the sound of laptop speaker, and the irony is expressed by contrasting two situations, which are “quiet for music” and “loud for porn”. Each of these two phrases does not have the explicit sentiment until they are in a contrasting context, and our model has successfully identified most key patterns for building the implicit context incongruity. The last example has two named entities as the context incongruity, which does not really have sentiment meaning until our model pass the learned sentiment knowledge to them. Both “Justin Bieber” and “philosophy professor” appear a few times in our sentiment training corpus, and tweets with “Justin Bieber” are more likely to be negative while tweets with “philosophy professor” are more likely to be positive. Even though both named entities do not have sentiment meaning in general, the supervised sentiment training can embed an implicit sentiment via deep features. With the implicit sentiment features learned in sentiment training, our irony model STBi-LSTM successfully detects the context incongruity at the second stage of learning.

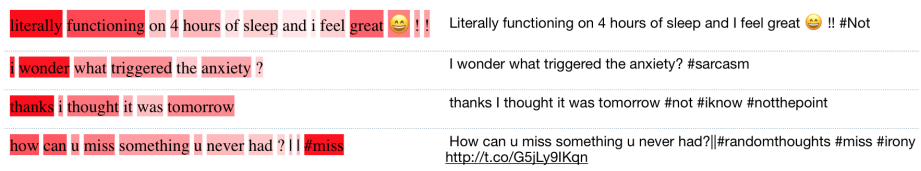

Figure 7: Examples of mistakenly classified tweets (ironic tweets have been classified as non-ironic): tweets in the left column are chosen from the SemEval2018 test data, tweets in the right column are their original version where all hashtags are kept. (The luminance of red represents the attention value of each word paid by our STBi-LSTM model).

6.4. Limitations

Our models can only detect irony based on the self-contained contents in tweets. In order to understand the what our models miss on irony detection, we provide several

mistakenly classified examples with their original version of text content in Fig. 7. In ironic tweets, hashtags such as #irony, #sarcasm and #not are often used to indicate the irony intention. When these hashtags are removed for learning a more general model, it is hard to imagine that even humans can classify such tweets as ironic. For example, the second and third examples, none of them carry the irony sense when the hashtags "#sarcasm" and "not" are removed.

Some irony can only be inferred from the conversational context. As a result, when the complete conversational context is not available, it is even hard for a human to find the irony utterance [9, 16]. In our examples, the second and third tweets are more likely coming from a conversational context where the authors of these tweets wrote them to express ironic intent. In the first example, our model successfully detected the positive sentiment "feel great", but failed to detect the negative situation "4 hours of sleep".

7. Conclusion

In this paper, we studied the problem of irony detection on Twitter. Context incongruity is a commonly seen form of irony on Twitter, where the contrast between the positive statement and the negative context is the common form of context incongruity. We proposed to employ transfer learning and attention-based neural network to identify context incongruity for detecting irony. The most challenging part for training a good automatic irony detection model is the limited human labeled dataset. In contrast with irony detection, sentiment analysis has sufficient resources, such as predefined sentiment lexica and human annotated corpora. We proposed our models to take advantage of these widely and readily available sentiment resources to improve the ability of attention-based model on detecting context incongruity. With incorporating transferred sentiment, our models are able to detect both implicit and explicit context incongruity at most times. Experiments show that our three proposed sentiment attention mechanisms result in better performance than the baselines including several popular neural models for irony detection on Twitter.

References

[1] F. D. I. Hernańdez, V. Patti, P. Rosso, Irony detection in twitter: The role of affective content, ACM Transactions on Internet Technology 16 (3) (2016) 19.

[2] I. Hernández, R. Paolo, Irony, sarcasm, and sentiment analysis, in: F. A. Pozzi, E. Fersini, E. Messina, B. Liu (Eds.), Sentiment analysis in social networks, Morgan Kaufmann, 2016, Ch. 7, pp. 113–128.

[3] P. Rosso, F. Rangel, I. H. Farías, L. Cagnina, W. Zaghouani, A. Charfi, A survey on author profiling, deception, and irony detection for the arabic language, Language and Linguistics Compass 12 (4) (2018) e12275.

[4] R. González-Ibáñez, S. Muresan, N. Wacholder, Identifying sarcasm in twitter: A closer look, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 581–586.

[5] F. Barbieri, V. Basile, D. Croce, M. Nissim, N. Novielli, V. Patti, Overview of the evalita 2016 sentiment polarity classification task, in: Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), 2016.

[6] R. J. Gerrig, Y. Goldvarg, Additive effects in the perception of sarcasm: Situational disparity and echoic mention, Metaphor and Symbol 15 (4) (2000) 197–208.

[7] S. L. Ivanko, P. M. Pexman, Context incongruity and irony processing, Discourse Processes 35 (3) (2003) 241–279.

[8] A. Joshi, V. Sharma, P. Bhattacharyya, Harnessing context incongruity for sarcasm detection, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Vol. 2, 2015, pp. 757–762.

[9] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, R. Huang, Sarcasm as contrast between a positive sentiment and negative situation, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 704–714.

[10] A. Joshi, P. Bhattacharyya, M. J. Carman, Automatic sarcasm detection: A survey, ACM Computing Surveys (CSUR) 50 (5) (2017) 73.

[11] B. C. Wallace, Computational irony: A survey and new perspectives, Artificial Intelligence Review 43 (4) (2015) 467–483.

[12] S. Poria, E. Cambria, D. Hazarika, P. Vij, A deeper look into sarcastic tweets using deep convolutional neural networks, in: Proceedings of the 26th International Conference on Computational Linguistics, 2016, pp. 1601–1612.

[13] A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, M. Carman, Are word embedding-based features useful for sarcasm detection?, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1006–1011.

[14] Y.-H. Huang, H.-H. Huang, H.-H. Chen, Irony detection with attentive recurrent neural networks, in: Proceedings of European Conference on Information Retrieval, Springer, 2017, pp. 534–540.

[15] S. Oraby, V. Harrison, A. Misra, E. Riloff, M. Walker, Are you serious?: Rhetorical questions and sarcasm in social media dialog, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017, pp. 310–319.

[16] D. Ghosh, A. R. Fabbri, S. Muresan, The role of conversation context for sarcasm detection in online interactions, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017, pp. 186–196.

[17] A. Ghosh, T. Veale, Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 482–491.

[18] Y. Tay, L. A. Tuan, S. C. Hui, J. Su, Reasoning with sarcasm by reading in-between, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018.

[19] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural, 2005, p. 347.

[20] P. K. Novak, J. Smailović, B. Sluban, I. Mozetič, Sentiment of emojis, PloS one 10 (12) (2015) e0144296.

[21] D. Maynard, M. A. Greenwood, Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis., in: LREC, 2014, pp. 4238–4243.

[22] E. Sulis, D. I. H. Farías, P. Rosso, V. Patti, G. Ruffo, Figurative messages and affect in twitter: Differences between# irony,# sarcasm and# not, Knowledge-Based Systems 108 (2016) 132–143.

[23] F. Kunneman, C. Liebrecht, M. Van Mulken, A. Van den Bosch, Signaling sarcasm: From hyperbole to hashtag, Information Processing & Management 51 (4) (2015) 500–509.

[24] A. Ghosh, T. Veale, Fracking sarcasm using neural network, in: Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2016, pp. 161–169.

[25] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, S. Lehmann, Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1615–1625.

[26] Y. Wang, M. Huang, L. Zhao, et al., Attention-based lstm for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.

[27] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Vol. 2, 2016, pp. 207–212.

[28] T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.

[29] R. Xia, F. Xu, J. Yu, Y. Qi, E. Cambria, Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis, Information Processing & Management 52 (1) (2016) 36–45.

[30] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[31] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, 2012, pp. 17–36.

[32] K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, Journal of Big Data 3 (1) (2016) 9.

[33] Z. Cao, W. Li, S. Li, F. Wei, Improving multi-document summarization via text classification, in: Proceedings of AAAI, 2017, pp. 3053–3059.

[34] X. Shu, G.-J. Qi, J. Tang, J. Wang, Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation, in: Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 35–44.

[35] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining., in: LREC, Vol. 10, 2010, pp. 2200–2204.

[36] C. Hutto, E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of Eighth International AAAI Conference on Weblogs and Social Media, 2014.

[37] S. Rosenthal, N. Farra, P. Nakov, Semeval-2017 task 4: Sentiment analysis in twitter, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 502–518.

[38] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, A. Reyes, Semeval-2015 task 11: Sentiment analysis of figurative language in twitter, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 470–478.

[39] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751.

[40] A. Reyes, P. Rosso, T. Veale, A multidimensional approach for detecting irony in twitter, Language resources and evaluation 47 (1) (2013) 239–268.

[41] F. Barbieri, H. Saggion, F. Ronzano, Modelling sarcasm in twitter, a novel approach, in: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2014, pp. 50–58.

[42] T. Ptáček, I. Habernal, J. Hong, Sarcasm detection on czech and english twitter, in: Proceedings of the 25th International Conference on Computational Linguistics, 2014, pp. 213–223.

[43] S. M. Mohammad, X. Zhu, S. Kiritchenko, J. Martin, Sentiment, emotion, purpose, and style in electoral tweets, Information Processing & Management 51 (4) (2015) 480–499.

[44] C. Van Hee, E. Lefever, V. Hoste, Semeval-2018 task 3: Irony detection in english tweets, in: Proceedings of The 12th International Workshop on Semantic Evaluation, 2018, pp. 39–50.

[45] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing, 2014, pp. 1532–1543.