SemEval-2018 Task 3: Irony Detection in English Tweets

Cynthia Van Hee, Els Lefever and **Véronique Hoste**
LT3 Language and Translation Technology Team
Ghent University
Groot-Brittanniëlaan 45, 9000 Ghent
firstname.lastname@ugent.be

Abstract

This paper presents the first shared task on irony detection: given a tweet, automatic natural language processing systems should determine whether the tweet is ironic (Task A) and which type of irony (if any) is expressed (Task B). The ironic tweets were collected using irony-related hashtags (i.e. *#irony, #sarcasm, #not*) and were subsequently manually annotated to minimise the amount of noise in the corpus. Prior to distributing the data, hashtags that were used to collect the tweets were removed from the corpus. For both tasks, a training corpus of 3,834 tweets was provided, as well as a test set containing 784 tweets. Our shared tasks received submissions from 43 teams for the binary classification Task A and from 31 teams for the multiclass Task B. The highest classification scores obtained for both subtasks are respectively F_1= 0.71 and F_1= 0.51 and demonstrate that fine-grained irony classification is much more challenging than binary irony detection.

1 Introduction

The development of the social web has stimulated the use of figurative and creative language, including irony, in public (Ghosh et al., 2015). From a philosophical/psychological perspective, discerning the mechanisms that underlie ironic speech improves our understanding of human reasoning and communication, and more and more, this interest in understanding irony also emerges in the machine learning community (Wallace, 2015). Although an unanimous definition of irony is still lacking in the literature, it is often identified as a trope whose actual meaning differs from what is literally enunciated. Due to its nature, irony has important implications for natural language processing (NLP) tasks, which aim to understand and produce human language. In fact, automatic irony detection has a large potential for various applications in the domain of text mining, especially those that require semantic analysis, such as author profiling, detecting online harassment, and, maybe the most well-known example, sentiment analysis.

Due to its importance in industry, sentiment analysis research is abundant and significant progress has been made in the field (e.g. in the context of SemEval (Rosenthal et al., 2017)). However, the SemEval-2014 shared task *Sentiment Analysis in Twitter* (Rosenthal et al., 2014) demonstrated the impact of irony on automatic sentiment classification by including a test set of ironic tweets. The results revealed that, while sentiment classification performance on regular tweets reached up to F_1= 0.71, scores on the ironic tweets varied between F_1= 0.29 and F_1= 0.57. In fact, it has been demonstrated that several applications struggle to maintain high performance when applied to ironic text (e.g. Liu, 2012; Maynard and Greenwood, 2014; Ghosh and Veale, 2016). Like other types of figurative language, ironic text should not be interpreted in its literal sense; it requires a more complex understanding based on associations with the context or world knowledge. Examples 1 and 2 are sentences that regular sentiment analysis systems would probably classify as positive, whereas the intended sentiment is undeniably negative.

(1) *I feel so blessed to get ocular migraines.*

(2) *Go ahead drop me hate, I'm looking forward to it.*

For human readers, it is clear that the author of example 1 does not feel blessed at all, which can be inferred from the contrast between the positive sentiment expression "I feel so blessed", and the negative connotation associated with getting ocular migraines. Although such connotative infor-

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 39–50
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

mation is easily understood by most people, it is difficult to access by machines. Example 2 illustrates implicit cyberbullying; instances that typically lack explicit profane words and where the offense is often made through irony. Similarly to example 1, a contrast can be perceived between a positive statement ("I'm looking forward to") and a negative situation (i.e. experiencing hate). To be able to interpret the above examples correctly, machines need, similarly to humans, to be aware that irony is used, and that the intended sentiment is opposite to what is literally enunciated.

The irony detection task[1] we propose is formulated as follows: given a single post (i.e. a tweet), participants are challenged to automatically determine whether irony is used and which type of irony is expressed. We thus defined two subtasks:

- Task A describes a **binary irony classification task** to define, for a given tweet, whether irony is expressed.
- Task B describes a **multiclass irony classification task** to define whether it contains a specific type of irony (verbal irony by means of a polarity clash, situational irony, or another type of verbal irony, see further) or is not ironic. Concretely, participants should define which one out of four categories a tweet contains: ironic by clash, situational irony, other verbal irony or not ironic.

It is important to note that by a tweet, we understand the actual text it contains, without metadata (e.g. user id, time stamp, location). Although such metadata could help to recognise irony, the objective of this task is to learn, at message level, how irony is linguistically realised.

2 Automatic Irony Detection

As described by Joshi et al. (2017), recent approaches to irony can roughly be classified as either rule-based or (supervised and unsupervised) machine learning-based. While rule-based approaches mostly rely upon lexical information and require no training, machine learning invariably makes use of training data and exploits different types of information sources (or *features*), such as bags of words, syntactic patterns, sentiment information or semantic relatedness.

Previous work on irony detection mostly applied supervised machine learning mainly exploiting lexical features. Other features often include punctuation mark/interjection counts (e.g Davidov et al., 2010), sentiment lexicon scores (e.g. Bouazizi and Ohtsuki, 2016; Farías et al., 2016), emoji (e.g. González-Ibáñez et al., 2011), writing style, emotional scenarios, part of speech-patterns (e.g. Reyes et al., 2013), and so on. Also beneficial for this task are combinations of different feature types (e.g. Van Hee et al., 2016b), author information (e.g. Bamman and Smith, 2015), features based on (semantic or factual) oppositions (e.g Karoui et al., 2015; Gupta and Yang, 2017; Van Hee, 2017) and even eye-movement patterns of human readers (Mishra et al., 2016). While a wide range of features are and have been used extensively over the past years, deep learning techniques have recently gained increasing popularity for this task. Such systems often rely on semantic relatedness (i.e. through word and character embeddings (e.g. Amir et al., 2016; Ghosh and Veale, 2016)) deduced by the network and reduce feature engineering efforts.

Regardless of the methodology and algorithm used, irony detection often involves binary classification where irony is defined as instances that express the opposite of what is meant (e.g. Riloff et al., 2013; Joshi et al., 2017). Twitter has been a popular data genre for this task, as it is easily accessible and provides a rapid and convenient method to find (potentially) ironic messages by looking for hashtags like *#irony, #not* and *#sarcasm.* As a consequence, irony detection research often relies on automatically annotated (i.e. based on irony-related hashtags) corpora, which contain noise (Kunneman et al., 2015; Van Hee, 2017).

3 Task Description

We propose two subtasks A and B for the automatic detection of irony on Twitter, for which we provide more details below.

3.1 Task A: Binary Irony Classification

The first subtask is a two-class (or binary) classification task where submitted systems have to predict whether a tweet is ironic or not. The following examples respectively present an ironic and non-ironic tweet.

(3) *I just love when you test my patience!! #not.*

[1] All practical information, data download links and the final results can be consulted via the CodaLab website of our task: https://competitions.codalab.org/competitions/17468.

(4) *Had no sleep and have got school now #not happy*

Note that the examples contain irony-related hashtags (e.g. #irony) that were removed from the corpus prior to distributing the data for the task.

3.2 Task B: Multiclass Irony Classification

The second subtask is a multiclass classification task where submitted systems have to predict one out of four labels describing i) verbal irony realised through a polarity contrast, ii) verbal irony without such a polarity contrast (i.e. other verbal irony), iii) descriptions of situational irony, and iv) non-irony. The following paragraphs present a description and a number of examples for each label.

Verbal irony by means of a polarity contrast This category applies to instances containing an evaluative expression whose polarity (positive, negative) is inverted between the literal and the intended evaluation, as shown in examples 5 and 6:

(5) *I love waking up with migraines #not* 😫

(6) *I really love this year's summer; weeks and weeks of awful weather*

In the above examples, the irony results from a polarity inversion between two evaluations. For instance, in example 6, the literal evaluation ("I really love this year's summer") is positive, while the intended one, which is implied by the context ("weeks and weeks of awful weather"), is negative.

Other verbal irony This category contains instances that show no polarity contrast between the literal and the intended evaluation, but are nevertheless ironic.

(7) *@someuser Yeah keeping cricket clean, that's what he wants #Sarcasm*

(8) *Human brains disappear every day. Some of them have never even appeared. http://t.co/Fb0Aq5Frqs #brain #humanbrain #Sarcasm*

Situational irony This class label is reserved for instances describing situational irony, or situations that fail to meet some expectations. As explained by Shelley (2001), firefighters who have a fire in their kitchen while they are out to answer a fire alarm would be a typically ironic situation. Some other examples of situational irony are the following:

(9) *Most of us didn't focus in the #ADHD lecture. #irony*

(10) *Event technology session is having Internet problems. #irony #HSC2024*

Non-ironic This class contains instances that are clearly not ironic, or which lack context to be sure that they are ironic, as shown in the following examples:

(11) *And then my sister should be home from college by time I get home from babysitting. And it's payday. THIS IS A GOOD FRIDAY*

(12) *Is Obamacare Slowing Health Care Spending? #NOT*

4 Corpus Construction and Annotation

A data set of 3,000 English tweets was constructed by searching Twitter for the hashtags *#irony, #sarcasm* and *#not* (hereafter referred to as the 'hashtag corpus'), which could occur anywhere in the tweet that was finally included in the corpus. All tweets were collected between 01/12/2014 and 04/01/2015 and represent 2,676 unique users. To minimise the noise introduced by groundless irony hashtags, all tweets were manually labelled using a fine-grained annotation scheme for irony (Van Hee et al., 2016a). Prior to data annotation, the entire corpus was cleaned by removing retweets, duplicates and non-English tweets and replacing XML-escaped characters (e.g. `&amp;`).

The corpus was entirely annotated by three students in linguistics and second-language speakers of English, with each student annotating one third of the whole corpus. All annotations were done using the brat rapid annotation tool (Stenetorp et al., 2012). To assess the reliability of the annotations, and whether the guidelines allowed to carry out the task consistently, an **interannotator agreement study** was set up in two rounds. Firstly, inter-rater agreement was calculated between the authors of the guidelines to test the guidelines for usability and to assess whether changes or additional clarifications were recommended prior annotating the entire corpus. For this purpose, a subset of 100 instances from the SemEval-2015 Task *Sentiment Analysis of Figurative Language in Twitter* (Ghosh et al., 2015) dataset were annotated. Based on the results, some clarifications and refinements were added to the annotation scheme, which are thoroughly described in Van Hee (2017). Next, a second agreement study was carried out on a subset (i.e. 100 randomly chosen instances) of the corpus. As metric, we used **Fleiss' Kappa** (Fleiss, 1971), a widespread statistical measure in the field of computational linguistics for assessing annotator agreement on categorical ratings (Carletta, 1996). The measure calculates the degree of agreement in classification over the agreement which would be expected by chance, i.e. when annotators would randomly assign class labels.

annotation	**Kappa κ round 1**	**Kappa κ round 2**
ironic / not ironic	0.65	0.72
ironic by clash / other / not ironic	0.55	0.72

Table 1: Inter-annotator agreement scores (Kappa) in two annotation rounds.

Table 1 presents the inter-rater scores for the binary irony distinction and for three-way irony classification ('other' includes both situational irony and other forms of verbal irony). We see that better inter-annotator agreement is obtained after the refinement of the annotation scheme, especially for the binary irony distinction. Given the difficulty of the task, a Kappa score of 0.72 for recognising irony can be interpreted as good reliability[2].

The distribution if the different irony types in the experimental corpus are presented in Table 2.

class label	**# instances**
Verbal irony by means of a polarity contrast	1,728
Other types of verbal irony	267
Situational irony	401
Non-ironic	604

Table 2: Distribution of the different irony categories in the corpus

Based on the annotations, 2,396 instances out of the 3,000 are ironic, while 604 are not. To balance the class distribution in our experimental corpus, 1,792 non-ironic tweets were added from a background corpus. The tweets in this corpus were collected from the same set of Twitter users as in the hashtag corpus, and within the same time span. It is important to note that these tweets do not contain irony-related hashtags (as opposed to the non-ironic tweets in the hashtag corpus), and were manually filtered from ironic tweets. Adding these non-ironic tweets to the experimental corpus brought the total amount of data to 4,792 tweets (2,396 ironic + 2,396 non-ironic). For this shared task, the corpus was randomly split into a class-balanced training (80% or 3,833 instances) and test (20%, or 958 instances) set. In an additional cleaning step, we removed ambiguous tweets (i.e. where additional context was required to understand their ironic nature), from the test corpus, resulting in a test set containing 784 tweets (consisting of 40% ironic and 60% non-ironic tweets).

To train their systems, participants were not restricted to the provided training corpus. They were allowed to use additional training data that was collected and annotated at their own initiative. In the latter case, the submitted system was considered *unconstrained*, as opposed to *constrained* if only the distributed training data were used for training.

It is important to note that participating teams were allowed ten submissions at CodaLab, and that they could submit a constrained and unconstrained system for each subtask. However, only their last submission was considered for the official ranking (see Table 3).

5 Evaluation

For both subtasks, participating systems were evaluated using standard evaluation metrics, including accuracy, precision, recall and F$_1$ score, calculated as follows:

$$accuracy = \frac{true\,positives\ +\ true\,negatives}{total\ number\ of\ instances} \tag{1}$$

$$precision = \frac{true\,positives}{true\,positives\ +\ false\,positives} \tag{2}$$

$$recall = \frac{true\,positives}{true\,positives\ +\ false\,negatives} \tag{3}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision\ +\ recall} \tag{4}$$

While accuracy provides insights into the system performance for all classes, the latter three measures were calculated for the positive class only (Task A) or were macro-averaged over four class labels (Task B). Macro-averaging of the F$_1$ score implies that all class labels have equal weight in the final score.

[2] According to magnitude guidelines by Landis and Koch (1977).

For both subtasks, two baselines were provided against which to compare the systems' performance. The first baseline randomly assigns irony labels and the second one is a linear SVM classifier with standard hyperparameter settings exploiting tf-idf word unigram features (implemented with scikit-learn (Pedregosa et al., 2011)). The second baseline system is made available to the task participants via GitHub[3].

6 Systems and results for Task A

In total, 43 teams competed in Task A on binary irony classification. Table 3 presents each team's performance in terms of accuracy, precision, recall and F_1 score. In all tables, the systems are ranked by the official F_1 score (shown in the fifth column). Scores from teams that are marked with an asterisk should be interpreted carefully, as the number of predictions they submitted does not correspond to the number of test instances.

As can be observed from the table, the SVM unigram baseline clearly outperforms the random class baseline and generally performs well for the task. Below we discuss the top five best-performing teams for Task A, which all built a constrained (i.e. only the provided training data were used) system. The best system yielded an F_1 score of 0.705 and was developed by THU_NGN (Wu et al., 2018). Their architecture consists of densely connected LSTMs based on (pre-trained) word embeddings, sentiment features using the AffectiveTweet package (Mohammad and Bravo-Marquez, 2017) and syntactic features (e.g. PoS-tag features + sentence embedding features). Hypothesising that the presence of a certain irony hashtag correlates with the type of irony that is used, they constructed a multi-task model able to predict simultaneously 1) the missing irony hashtag, 2) whether a tweet is ironic or not and 3) which fine-grained type of irony is used in a tweet.

Also in the top five are the teams NTUA-SLP (F_1= 0.672), WLV (F_1= 0.650), NLPRL-IITBHU (F_1= 0.648) and NIHRIO (F_1= 0.648). NTUA-SLP (Baziotis et al., 2018) built an ensemble classifier of two deep learning models: a word- and character-based (bi-directional) LSTM to capture semantic and syntactic information in tweets, respectively. As features, the team used pre-trained character and word embeddings on a corpus of 550 million tweets. Their ensemble classifier applied majority voting to combine the outcomes of the two models. WLV (Rohanian et al., 2018) developed an ensemble voting classifier with logistic regression (LR) and a support vector machine (SVM) as component models. They combined (through averaging) pre-trained word and emoji embeddings with handcrafted features, including sentiment contrasts between elements in a tweet (i.e. left vs. right sections, hashtags vs. text, emoji vs. text), sentiment intensity and word-based features like flooding and capitalisation). For Task B, they used a slightly altered (i.e. ensemble LR models and concatenated word embeddings instead of averaged) model. NLPRL-IITBHU (Rangwani et al., 2018) ranked fourth and used an XGBoost Classifier to tackle Task A. They combined pre-trained CNN activations using DeepMoji (Felbo et al., 2017) with ten types of handcrafted features. These were based on polarity contrast information, readability metrics, context incongruity, character flooding, punctuation counts, discourse markers/intensifiers/interjections/swear words counts, general token counts, WordNet similarity, polarity scores and URL counts. The fifth best system for Task A was built by NIHRIO (Vu et al., 2018) and consists of a neural-networks-based architecture (i.e. Multilayer Perceptron). The system exploited lexical (word- and character-level unigrams, bigrams and trigrams), syntactic (PoS-tags with tf-idf values), semantic features (word embeddings using GloVe (Pennington et al., 2014), LSI features and Brown cluster features (Brown et al., 1992)) and polarity features derived from the Hu and Liu Opinion Lexicon (Hu and Liu, 2004).

As such, all teams in the top five approached the task differently, by exploiting various algorithms and features, but all of them clearly outperformed the baselines. Like most other teams, they also showed a better performance in terms of recall compared to precision.

Table 3 displays the results of each team's official submission for Task A, i.e. no distinction is made between constrained and unconstrained systems. By contrast, Tables 4 and 5 present the rankings of the **best** (i.e. not necessarily the last, and hence official submission) constrained and unconstrained submissions for Task A.

As can be deduced from Table 4, when considering all constrained submissions from each team and ranking them based on performance, we see

[3] https://github.com/Cyvhee/SemEval2018-Task3/

team	acc	precision	recall	F_1
THU_NGN	**0.735**	0.630	0.801	**0.705**
NTUA-SLP	0.732	0.654	0.691	0.672
WLV	0.643	0.532	0.836	0.650
NLPRL-IITBHU	0.661	0.551	0.788	0.648
NIHRIO	0.702	0.609	0.691	0.648
DLUTNLP-1	0.628	0.520	0.797	0.629
ELiRF-UPV	0.611	0.506	0.833	0.629
liangxh16	0.659	0.555	0.714	0.625
CJ	0.667	0.565	0.695	0.623
#NonDicevo-SulSerio	0.679	0.583	0.666	0.622
UWB	0.688	0.599	0.643	0.620
INAOE-UPV	0.651	0.546	0.714	0.618
RM@IT	0.649	0.544	0.714	0.618
DUTQS	0.601	0.498	0.794	0.612
ISP RAS	0.565	0.473	**0.849**	0.608
ValenTO	0.598	0.496	0.781	0.607
[4]binarizer	0.666	0.553	0.647	0.596
SIRIUS_LC	0.684	0.604	0.588	0.596
warnikchow	0.644	0.543	0.656	0.594
ECNU	0.596	0.494	0.743	0.593
Parallel Computing-Network Research Group	0.617	0.513	0.701	0.592
Lancaster	0.635	0.532	0.666	0.591
Unigram SVM BL	0.635	0.532	0.659	0.589
IITBHU-NLP	0.566	0.472	0.778	0.587
s1998	0.629	0.526	0.653	0.583
Random Decision - Syntax Trees	0.617	0.514	0.672	0.582
textbflyreact	0.628	0.525	0.640	0.577
UTH-SU	0.639	0.540	0.605	0.571
KLUEnicorn	0.594	0.491	0.643	0.557
ai-ku	0.643	0.555	0.502	0.527
UTMN	0.603	0.500	0.556	0.527
UCDCC	0.682	0.645	0.444	0.526
IITG	0.556	0.450	0.540	0.491
MI&T–LAB	0.614	0.514	0.463	0.487
*NEUROSENT-PDI	0.504	0.409	0.560	0.472
Lovelace	0.512	0.412	0.543	0.469
codersTeam	0.509	0.410	0.543	0.468
WHLL	0.580	0.469	0.437	0.453
DKE_UM	0.561	0.447	0.450	0.449
LDR	0.564	0.446	0.415	0.430
*YNU-HPCC	0.509	0.391	0.428	0.408
Random BL	0.503	0.373	0.373	0.373
ACMK-POZNAN	0.620	0.550	0.232	0.326
iiidyt	0.352	0.257	0.334	0.291
milkstout	0.584	0.427	0.142	0.213
INGEOTEC-IIMAS	0.628	**0.880**	0.071	0.131

Table 3: Official (CodaLab) results for Task A, ranked by F_1 score. The highest scores in each column are shown in bold and the baselines are indicated in purple.

that the UCDCC team ranks first (F_1= 0.724), followed by THU_NGN, NTUA-SLP, WLV and NLPRL-IITBHU, whose approach was discussed earlier in this paper. The UCDCC-system is an LSTM model exploiting Glove word embedding features.

team	acc	precision	recall	F_1
UCDCC	**0.797**	**0.788**	0.669	**0.724**
THU_NGN	0.735	0.630	0.801	0.705
NTUA-SLP	0.732	0.654	0.691	0.672
WLV	0.643	0.532	**0.836**	0.650
NLPRL-IITBHU	0.661	0.551	0.788	0.648
NCL	0.702	0.609	0.691	0.648
RM@IT	0.691	0.598	0.679	0.636
#NonDicevo-SulSerio	0.666	0.562	0.717	0.630
DLUTNLP-1	0.628	0.520	0.797	0.629
ELiRF-UPV	0.611	0.506	0.833	0.629

Table 4: Best constrained systems for Task A.

team	acc	precision	recall	F_1
#NonDicevo-SulSerio	**0.679**	0.583	0.666	**0.622**
INAOE-UPV	0.651	0.546	**0.714**	0.618
RM@IT	0.649	0.544	0.714	0.618
ValenTO	0.598	0.496	**0.781**	0.607
UTMN	0.603	0.500	0.556	0.527
IITG	0.556	0.450	0.540	0.491
LDR	0.571	0.455	0.408	0.431
milkstouts	0.584	0.427	0.142	0.213
INGEOTEC-IIMAS	0.643	**0.897**	0.113	0.200

Table 5: Best unconstrained systems for Task A.

In the top five unconstrained (i.e. using additional training data) systems for Task A are #NonDicevoSulSerio, INAOE-UPV, RM@IT, ValenTO and UTMN, with F_1 scores ranging between 0.622 and 0.527. #NonDicevoSulserio extended the training corpus with 3,500 tweets from existing irony corpora (e.g. Riloff et al. (2013); Barbieri and Saggion (2014); Ptáček et al. (2014) and built an SVM classifier exploiting structural features (e.g. hashtag count, text length), sentiment- (e.g. contrast between text and emoji sentiment), and emotion-based (i.e. emotion lexicon scores) features. INAOE-UPV combined pretrained word embeddings from the Google News corpus with word-based features (e.g. n-grams). They also extended the official training data with benchmark corpora previously used in irony research and trained their system with a total of 165,000 instances. RM@IT approached the task using an ensemble classifier based on attention-based recurrent neural networks and the Fast-

Text (Joulin et al., 2017) library for learning word representations. They enriched the provided training corpus with, on the one hand, the data sets provided for SemEval-2015 Task 11 (Ghosh et al., 2015) and, on the other hand, the sarcasm corpus composed by Ptáček et al. (2014). Altogether, this generated a training corpus of approximately 110,000 tweets. ValenTO took advantage of irony corpora previously used in irony detection that were manually annotated or through crowdsourcing (e.g. Riloff et al., 2013; Ptáček et al., 2014). In addition, they extended their corpus with an unspecified number of self-collected irony tweets using the hashtags *#irony* and *#sarcasm*. Finally, UTMN developed an SVM classifier exploiting binary bag-of-words features. They enriched the training set with 1,000 humorous tweets from SemEval-2017 Task 6 (Potash et al., 2017) and another 1,000 tweets with positive polarity from SemEval-2016 Task 4 (Nakov et al., 2016), resulting in a training corpus of 5,834 tweets.

Interestingly, when comparing the best constrained with the best unconstrained system for Task A, we see a difference of 10 points in favour of the constrained system, which indicates that adding more training data does not necessarily improve the classification performance.

7 Systems and Results for Task B

While 43 teams competed in Task A, 31 teams submitted a system for Task B on multiclass irony classification. Table 6 presents the official ranking with each team's performance in terms of accuracy, precision, recall and F_1 score. Similar to Task A, we discuss the top five systems in the overall ranking (Table 6) and then zoom in on the best performing constrained and unconstrained systems (Tables 7 and 8).

For Task B, the top five is nearly similar to the top five for Task A and includes the following teams: UCDCC (Ghosh, 2018), NTUA-SLP (Baziotis et al., 2018), THU_NGN (Wu et al., 2018), NLPRL-IITBHU (Rangwani et al., 2018) and NIHRIO (Vu et al., 2018). All of the teams tackled multiclass irony classification by applying (mostly) the same architecture as for Task A (see earlier). Inspired by siamese networks (Bromley et al., 1993) used in image classification, the UCDCC team developed a siamese architecture for irony detection in both subtasks. The neural network architecture makes use of Glove word embeddings as features and creates two identical subnetworks that are each fed with different parts of a tweet. Under the premise that ironic statements are often characterised by a form of opposition or contrast, the architecture captures this incongruity between two parts in an ironic tweet.

team	**acc**	**precision**	**recall**	**F_1**
UCDCC	**0.732**	**0.577**	0.504	**0.507**
NTUA-SLP	0.652	0.496	0.512	0.496
THU_NGN	0.605	0.486	**0.541**	0.495
NLPRL-IITBHU	0.603	0.466	0.506	0.474
NIHRIO	0.659	0.545	0.448	0.444
Random Decision Syntax Trees	0.633	0.487	0.439	0.435
ELiRF-UPV	0.633	0.412	0.440	0.421
WLV	0.671	0.431	0.415	0.415
#NonDicevo-SulSerio	0.545	0.409	0.441	0.413
INGEOTEC-IIMAS	0.644	0.502	0.385	0.406
ai-ku	0.584	0.422	0.402	0.393
warnikchow	0.598	0.412	0.410	0.393
UWB	0.626	0.440	0.406	0.390
CJ	0.603	0.412	0.409	0.384
UTH-SU	0.551	0.383	0.399	0.376
s1998	0.568	0.338	0.374	0.352
ValenTO	0.560	0.353	0.352	0.352
RM@IT	0.542	0.377	0.371	0.350
Unigram SVM BL	0.569	0.416	0.364	0.341
SSN_MLRG1	0.573	0.348	0.361	0.334
Lancaster	0.606	0.280	0.359	0.313
Parallel Computing Network Research Group	0.416	0.406	0.353	0.310
codersTeam	0.492	0.300	0.311	0.301
KLUEnicorn	0.347	0.321	0.353	0.298
DKE_UM	0.432	0.318	0.305	0.298
IITG	0.486	0.336	0.291	0.278
Lovelace	0.434	0.294	0.282	0.276
*YNU-HPCC	0.533	0.438	0.267	0.261
Random BL	0.416	0.241	0.241	0.241
LDR	0.461	0.230	0.250	0.234
ECNU	0.304	0.255	0.249	0.233
NEUROSENT-PDI	0.441	0.213	0.231	0.219
INAOE-UPV	0.594	0.217	0.261	0.215

Table 6: Official (CodaLab) results for Task B, ranked by F_1 score. The highest scores in each column are shown in bold and the baselines are indicated in purple.

NTUA-SLP, THU_NGN and NIHRIO used the same system for both subtasks. NLPRL-IITBHU also used the same architecture, but given the data skew for Task B, they used SMOTE (Chawla et al., 2002) as an oversampling technique to make sure each irony class was equally represented in the training corpus, which lead to an F_1 score increase of 5 points.

NLPRL-IITBHU built a Random Forest classifier making use of pre-trained DeepMoji embeddings, character embeddings (using Tweet2Vec) and sentiment lexicon features.

team	acc	precision	recall	F_1
UCDCC	**0.732**	**0.577**	0.504	**0.507**
NTUA-SLP	0.652	0.496	0.512	0.496
THU_NGN	0.605	0.486	**0.541**	0.495
NLPRL-IITBHU	0.603	0.466	0.506	0.474
NCL	0.659	0.545	0.448	0.444
Random Decision-Syntax Trees	0.633	0.487	0.439	0.435
ELiRF-UPV	0.633	0.412	0.440	0.421
WLV	0.671	0.431	0.415	0.415
AI-KU	0.584	0.422	0.402	0.393

Table 7: Best constrained systems for Task B. The highest scores in each column are shown in bold.

team	acc	precision	recall	F_1
#NonDicevo SulSerio	0.545	0.409	**0.441**	0.413
INGEOTEC-IIMAS	**0.647**	**0.508**	0.386	**0.407**
INAOE-UPV	0.495	0.347	0.379	0.350
IITG	0.486	0.336	0.291	0.278

Table 8: Unconstrained systems for Task B. The highest scores in each column are shown in bold.

As can be deduced from Table 7, the top five constrained systems correspond to the five best-performing systems overall (Table 6). Only four unconstrained systems were submitted for Task B. Differently from their Task A submission, #NonDicevoSulSerio applied a cascaded approach for this task, i.e. the first algorithm served an ironic/non-ironic classification, followed by a system distinguishing between ironic by clash and other forms of irony. Lastly, a third classifier distinguished between situational and other verbal irony. To account for class imbalance in step two, the team added 869 tweets of the *situational* and *other verbal irony* categories. INAOE-UPV, INGEOTEC-IIMAS and IITG also added tweets to the original training corpus, but it is not entirely clear how many were added and how these extra tweets were annotated.

Similar to Task A, the unconstrained systems do not seem to benefit from additional data, as they do not outperform the constrained submissions for the task.

team	not ironic	ironic by clash	situat. irony	other irony
UCDCC	**0.843**	**0.697**	0.376	0.114
NTUA-SLP	0.742	0.648	**0.460**	0.133
THU_NGN	0.704	0.608	0.433	**0.233**
NLPRL-IITBHU	0.689	0.636	0.387	0.185
NIHRIO	0.763	0.607	0.317	0.087
Random Decision-Syntax Trees	0.742	0.569	0.346	0.085
ELiRF-UPV	0.740	0.298	0.347	0.000
WLV	0.789	0.578	0.294	0.000
#NonDicevo SulSerio	0.683	0.533	0.315	0.121
INGEOTEC-IIMAS	0.764	0.494	0.211	0.152
ai-ku	0.699	0.529	0.258	0.087
warnikchow	0.717	0.524	0.300	0.028
UWB	0.744	0.557	0.232	0.027
CJ	0.724	0.559	0.202	0.050
*UTH-SU	0.671	0.513	0.254	0.065
s1998	0.711	0.446	0.253	0.000
emotIDM	0.713	0.456	0.165	0.074
RM@IT	0.671	0.481	0.148	0.100
SSN_MLRG1	0.704	0.499	0.105	0.027
Lancaster	0.729	0.523	0.000	0.000
Parallel Computing Network Res. Group	0.547	0.472	0.084	0.137
codersTeam	0.646	0.387	0.134	0.039
KLUEnicorn	0.423	0.384	0.200	0.186
DKE_UM	0.582	0.299	0.143	0.168
IITG	0.641	0.319	0.095	0.056
Lovelace	0.577	0.306	0.159	0.060
*YNU-HPCC	0.700	0.176	0.075	0.091
LDR	0.632	0.255	0.051	0.000
ECNU	0.444	0.259	0.118	0.110
*NEUROSENT-PDI	0.612	0.201	0.062	0.000
INAOE-UPV	0.748	0.000	0.111	0.000

Table 9: Results for Task B, reporting the F_1 score for the class labels. The highest scores in each column are shown in bold.

A closer look at the best and worst-performing systems for each subtask reveals that Task A benefits from systems that exploit a variety of handcrafted features, especially sentiment-based (e.g. sentiment lexicon values, polarity contrast), but also bags of words, semantic cluster features and PoS-based features. Other promising features for the task are word embeddings trained on large Twitter corpora (e.g. 5M tweets). The classifiers and algorithms used are (bidirectional) LSTMs, Random Forest, Multilayer Perceptron, and an optimised (i.e. using feature selection) voting classifier combining Support Vector Machines with Logistic Regression. Neural network-based systems exploiting word embeddings derived from the training dataset or generated from Wikipedia corpora perform less well for the task.

Similarly, Task B seems to benefit from (ensemble) neural-network architectures exploiting large corpus-based word embeddings and sentiment features. Oversampling and adjusting class weights are used to overcome the class imbalance of labels 2 and 3 versus 1 and 0 and tend to improve the classification performance. Ensemble classifiers outperform multi-step approaches and combined binary classifiers for this task.

Task B challenged the participants to distinguish between different types of irony. The class distributions in the training and test corpus are natural (i.e. no additional data were added after the annotation process) and imbalanced. For the evaluation of the task, F_1 scores were macro-averaged; on the one hand, this gives each label equal weight in the evaluation, but on the other hand, it does not show each class contribution to the average score. Table 9 therefore presents the participating teams' performance on each of the subtypes of irony in Task B. As can be deduced from Table 9, all teams performed best on the *non ironic* and *ironic by clash* classes, while identifying *situational irony* and *other irony* seems to be much more challenging. Although the scores for these two classes are the lowest, we observe an important difference between *situational* and *other* verbal irony. This can probably be explained by the heterogeneous nature of the *other* category, which collects diverse realisations of verbal irony. A careful and manual annotation of this class, which is currently being conducted, should provide more detailed insights into this category of ironic tweets.

8 Conclusions

The systems that were submitted for both subtasks represent a variety of neural-network-based approaches (i.e. CNNs, RNNs and (bi-)LSTMs) exploiting word- and character embeddings as well as handcrafted features. Other popular classification algorithms include Support Vector Machines, Maximum Entropy, Random Forest, and Naïve Bayes. While most approaches were based on one algorithm, some participants experimented with ensemble learners (e.g. SVM + LR, CNN + bi-LSTM, stacked LSTMs), implemented a voting system or built a cascaded architecture (for Task B) that first distinguished ironic from non-ironic tweets and subsequently differentiated between the fine-grained irony categories.

Among the most frequently used features are lexical features (e.g. n-grams, punctuation and hashtag counts, emoji presence) and sentiment- or emotion- lexicon features (e.g. based on SenticNet (Cambria et al., 2016), VADER (Hutto and Gilbert, 2014), aFinn (Nielsen, 2011)). Also important but to a lesser extent were syntactic (e.g. PoS-patterns) and semantic features, based on word, character and emoji embeddings or semantic clusters.

The best systems for Task A and Task B obtained an F_1 score of respectively 0.705 and 0.507 and clearly outperformed the baselines provided for this task. When looking at the scores per class label in Task B, we observe that high scores were obtained for the *non-ironic* and *ironic by clash* classes, and that *other irony* appears to be the most challenging irony type. Among all submissions, a wide variety of preprocessing tools, machine learning libraries and lexicons were explored.

As the provided datasets were relatively small, participants were allowed to include additional training data for both subtasks. Nevertheless, most submissions were constrained (i.e. only the provided training data were used): only nine unconstrained submissions were made for Task A, and four for Task B. When comparing constrained to unconstrained systems, it can be observed that adding more training data does not necessarily benefit the classification results. A possible explanation for this is that most unconstrained systems added training data from related irony research that were annotated differently (e.g. automatically) than the distributed corpus, which presumably limited the beneficial effect of increasing the training corpus size.

This paper provides some general insights into the main methodologies and bottlenecks for binary and multiclass irony classification. We observed that, overall, systems performed much better on Task A than Task B and the classification results for the subtypes of irony indicate that ironic by clash is most easily recognised (top F_1= 0.697), while other types of verbal irony and situational irony are much harder (top F_1 scores are 0.114 and 0.376, respectively).

References

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. *CoRR*, abs/1607.00976.

David Bamman and Noah A. Smith. 2015. Contextualized Sarcasm Detection on Twitter. In *Proceedings of the Ninth International Conference on Web and Social Media (ICWSM'15)*, pages 574–577, Oxford, UK. AAAI.

Francesco Barbieri and Horacio Saggion. 2014. Modelling Irony in Twitter. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the ACL*, pages 56–64, Gothenburg, Sweden. ACL.

Christos Baziotis, Nikolaos Athanasiou, Pinelopi Papalampidi, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, and Alexandros Potamianos. 2018. NTUA-SLP at SemEval-2018 Task 3: Deep Character and Word-level RNNs with Attention for Irony Detection in Twitter. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA. ACL.

Mondher Bouazizi and Tomoaki Ohtsuki. 2016. Sarcasm detection in twitter: "all your products are incredibly amazing!!!" - are they really? In *Global Communications Conference, GLOBECOM 2015*, pages 1–6. IEEE.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based N-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Bjoern Schuller. 2016. SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives. In *Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, pages 2666–2677, Osaka, Japan. ACL.

Jean Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16(1):321–357.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL'10)*, pages 107–116, Uppsala, Sweden. ACL.

Delia Irazú Hernańdez Farías, Viviana Patti, and Paolo Rosso. 2016. Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology*, 16(3):19:1–19:24.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. 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*, pages 1615–1625, Copenhagen, Denmark. ACL.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Aniruddha Ghosh. 2018. IronyMagnet at SemEval-2018 Task 3: A Siamese network for Irony detection in Social media. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA. ACL.

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. 2015. SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478, Denver, Colorado. ACL.

Aniruddha Ghosh and Tony Veale. 2016. Fracking Sarcasm using Neural Network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. ACL.

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 ACL: Human Language Technologies (HLT'11)*, pages 581–586, Portland, Oregon. ACL.

Raj Kumar Gupta and Yinping Yang. 2017. CrystalNest at SemEval-2017 Task 4: Using Sarcasm Detection for Enhancing Sentiment Classification and Quantification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 626–633. ACL.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Clayton J. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In *Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM-14)*, pages 216–225. AAAI.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic Sarcasm Detection:A Survey. *ACM Computing Surveys (CSUR)*, 50(5):73:1–73:22.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. ACL.

Jihen Karoui, Benamara Farah, Véronique MORICEAU, Nathalie Aussenac-Gilles, and Lamia Hadrich-Belguith. 2015. Towards a Contextual Pragmatic Model to Detect Irony in Tweets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 644–650, Beijing, China. ACL.

Florian Kunneman, Christine Liebrecht, Margot van Mulken, and Antal van den Bosch. 2015. Signaling sarcasm: From hyperbole to hashtag. *Information Processing Management*, 51(4):500–509.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1).

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Diana Maynard and Mark Greenwood. 2014. Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4238–4243, Reykjavik, Iceland. European Language Resources Association.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016. Harnessing Cognitive Features for Sarcasm Detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Berlin, Germany. ACL.

Saif Mohammad and Felipe Bravo-Marquez. 2017. Emotion Intensities in Tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics, *SEM @ACM 2017*, pages 65–77.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18, San Diego, California. ACL.

Finn Årup Nielsen. 2011. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, volume 718, pages 93–98.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. ACL.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. SemEval-2017 Task 6: #HashtagWars: Learning a Sense of Humor. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 49–57. ACL.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223, Dublin, Ireland. Dublin City University and ACL.

Harsh Rangwani, Devang Kulshreshtha, and Anil Kumar Sing. 2018. NLPRL-IITBHU at SemEval-2018 Task 3: Combining Linguistic Features and Emoji pre-trained CNN for Irony Detection in Tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA. ACL.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A Multidimensional Approach for Detecting Irony in Twitter. *Language Resources and Evaluation*, 47(1):239–268.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, pages 704–714, Seattle, Washington, USA. ACL.

Omid Rohanian, Shiva Taslimipoor, Richard Evans, and Ruslan Mitkov. 2018. WLV at SemEval-2018 Task 3: Dissecting Tweets in Search of Irony. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA. ACL.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. ACL.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of the*

8th International Workshop on Semantic Evaluation (SemEval 2014), pages 73–80, Dublin, Ireland. ACL and Dublin City University.

Cameron Shelley. 2001. The bicoherence theory of situational irony. *Cognitive Science*, 25(5):775–818.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: A Web-based Tool for NLP-assisted Text Annotation. In *Proceedings of the 13th Conference of the European Chapter of the ACL*, EACL'12, pages 102–107, Avignon, France. ACL.

Cynthia Van Hee. 2017. *Can machines sense irony? Exploring automatic irony detection on social media*. Ph.D. thesis, Ghent University.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2016a. Guidelines for Annotating Irony in Social Media Text, version 2.0. Technical Report 16-01, LT3, Language and Translation Technology Team–Ghent University.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2016b. Monday mornings are my fave #not: Exploring the Automatic Recognition of Irony in English tweets. In *Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, pages 2730–2739, Osaka, Japan.

Thanh Vu, Dat Quoc Nguyen, Xuan-Son Vu, Dai Quoc Nguyen, Michael Catt, and Michael Trenell. 2018. NIHRIO at SemEval-2018 Task 3: A Simple and Accurate Neural Network Model for Irony Detection in Twitter. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA. ACL.

Byron C. Wallace. 2015. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4):467–483.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. THU_NGN at SemEval-2018 Task 3: Tweet Irony Detection with Densely Connected LSTM and Multi-task Learning. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA. ACL.