UWB at SemEval-2018 Task 3: Irony detection in English tweets

Tomáš Hercig[1,2]

[1]NTIS – New Technologies for the Information Society,
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
[2]Department of Computer Science and Engineering,
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
tigi@kiv.zcu.cz
http://nlp.kiv.zcu.cz

Abstract

This paper describes our system created for the SemEval-2018 Task 3: Irony detection in English tweets.

Our strongly constrained system uses only the provided training data without any additional external resources. Our system is based on Maximum Entropy classifier and various features using parse tree, POS tags, and morphological features. Even without additional lexicons and word embeddings we achieved fourth place in Subtask A and seventh in Subtask B in terms of accuracy.

1 Introduction

Frequent use of creative and figurative language on social media has important implications for natural language processing tasks such as sentiment analysis. The semantics of a sentence with creative or figurative language can be quite different from the same sentence with literal meaning and misinterpreting figurative language such as irony represents a significant challenge in sentiment analysis. Hercig and Lenc (2017) explored the effect of figurative language on sentiment analysis and confirmed that figurative language affects sentiment analysis.

The issue of automatic irony and/or sarcasm[1] detection has been addressed mostly in English, however there has been some research in other languages as well (e.g. Dutch (Liebrecht et al., 2013), Italian (Bosco et al., 2013), Brazilian Portuguese (Vanin et al., 2013), and Czech (Ptáček et al., 2014)).

2 Task

The goal of SemEval-2018 Task 3 (Van Hee et al., 2018) is to detect irony in English tweets. Subtask A detects just binary score for irony and Subtask B also detects more detailed types of irony (non-ironic, ironic by clash, situational irony, and other forms of verbal irony). These subtasks correspond to their respective phases (A and B). Data for subtask B were available only after phase A was finished.

At the evaluation time the following descriptions of the submitted system labels were given:

- **Constrained:** only the provided training data were used to develop the system
- **Unconstrained:** additional training data were used

Only after the end of the phase A we learned that constrained systems can make use of additional resources like lexicons, dictionaries, embeddings, etc. Thus we introduce another system label to describe our system – strongly constrained.

- **Strongly Constrained:** using ONLY the the provided training/development data without any additional external resources (such as lexicons, embeddings, etc.)

Data statistics for Subtask A and Subtask B are shown in Table 1 and 2 respectively.

Label	Test	Train
Non-ironic	473 (60,3%)	1923 (50,2%)
Ironic	311 (39,7%)	1911 (49,8%)

Table 1: Data statistics for Subtask A.

3 System Description

For all experiments we use Maximum Entropy classifier with default settings from Brainy machine learning library (Konkol, 2014). Data preprocessing includes lower-casing and in some

[1] There is only a weak boundary in meaning between irony, sarcasm and satire (Reyes et al., 2012)

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 520–524
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

Label	Test	Train
Non-ironic	473 (60,3%)	1923 (50,2%)
Ironic by clash	164 (20,9%)	1390 (36,3%)
Situational irony	85 (10,8%)	316 (8,2%)
Other irony	62 (7,9%)	205 (5,3%)

Table 2: Data statistics for Subtask B.

cases lemmatization[2]. We utilize morphological analysis, parse trees, lemmatization, and POS tags from UDPipe (Straka et al., 2016).

3.1 Features

We tried to create the best strongly constrained feature set using various features using parse tree, POS tags, and morphological features. Most features listed below are based on the work of Hercig et al. (2016).

- **Character n-grams (ChN$_n$):** Separate binary feature for each n-gram representing the n-gram presence in the text. We do it separately for different orders $n \in \{1, 2, 3, 4, 5\}$ and remove n-gram with frequency $f \leq 2$.

- **Bag of Morphological features (BoM):** We use bag-of-words representation of a tweet, i.e. separate binary feature representing the occurrence of a morphological feature for all verbs in the tweet. The morphological features[3] include abbreviation, aspect, definiteness, degree of comparison, evidentiality, mood, polarity, politeness, possessive, pronominal type, tense, verb form, and voice.

- **Bag of Parse Tree Tags (BoT):** We use bag-of-words representation of a tweet, i.e. separate binary feature representing the occurrence of a parse tree tag in the tweet. We remove tags with a frequency $f \leq 2$.

- **First Words (FW):** Bag of first five words with at least 2 occurrences.

- **Last Words (LW):** Bag of last five words with at least 2 occurrences.

- **List (List):** Binary feature representing the presence of the following words or characters (yay, yep, yes, ha, heh, um, uh, sh, so, no, !, ?, .., ', ") in tweet.

- **N-gram Shape (NSh):** The occurrence of word shape n-gram in the tweet. Word shape assigns words into one of 24 classes[4] similar to the function specified in (Bikel et al., 1997). We consider unigrams with frequency $f > 2$ and trigrams with frequency $f > 10$.

- **POS Count (POS):** We use the count of POS tags in a tweet as a feature. We remove POS tags with frequency $f \leq 10$.

- **POS Count Bins (POS-B):** We map the frequency of POS tags in a tweet into a one-hot vector with length three and use this vector as binary features for the classifier. The frequency belongs to one of three equal-frequency bins[5]. Each bin corresponds to a position in the vector. We remove POS tags with frequency ≤ 5.

- **Root Bag of Words (R-BoW):** Bag of words for parent, siblings, and children of the root from the sentence parse tree. We use only words with POS[6] matching adjective, interjection, noun, symbol, verb, and other.

- **TF-IDF:** Term frequency – inverse document frequency of a word computed from the training data for words with at least 5 occurrences and at most 50 occurrences.

- **Verb Bag of Words (V-BoW):** Bag of words for parent, siblings, and children of the verb from the sentence parse tree. We use only words with POS[6] matching adverb, noun, adjective, verb, and auxiliary.

- **Word n-grams (WN$_n$):** Separate binary feature for each word n-gram representing the n-gram presence in the text. We do it separately for different orders $n \in \{1, 2, 3\}$ and remove n-gram with frequency $f \leq 2$.

3.2 Subtask A

We use a simple binary classification approach with the Maximum Entropy classifier and the features shown in Table 5. Blank space denotes that the corresponding feature has not been used.

[2] Character n-grams and N-gram Shape use original words.

[3] http://universaldependencies.org/u/feat/index.html

[4] We use edu.stanford.nlp.process.WordShapeClassifier with the WORDSHAPECHRIS1 setting available in Stanford CoreNLP library (Manning et al., 2014).

[5] The frequencies from the training data are split into three equal-size bins according to 33% quantiles.

[6] http://universaldependencies.org/u/pos/

Team	Accuracy	Precision	Recall	F1-score
THU_NGN	0.7347 (1)	0.6304 (4)	0.8006 (4)	0.7054 (1)
NTUA-SLP	0.7321 (2)	0.6535 (2)	0.6913 (13)	0.6719 (2)
NIHRIO, NCL	0.7015 (3)	0.6091 (5)	0.6913 (13)	0.6476 (5)
UWB best	0.7003	0.6195	0.6334	0.6264
UWB submitted	0.6875 (4)	0.5988 (7)	0.6431 (19)	0.6202 (11)

Table 3: CodaLab results for Subtask A.

Team	Accuracy	Precision	Recall	F1-score
UCDCC	0.7321 (1)	0.5768 (1)	0.5044 (4)	0.5074 (1)
WLV	0.6709 (2)	0.4311 (10)	0.4149 (9)	0.4153 (8)
NIHRIO, NCL	0.6594 (3)	0.5446 (2)	0.4475 (5)	0.4437 (5)
NTUA-SLP	0.6518 (4)	0.4959 (4)	0.5124 (2)	0.4959 (2)
INGEOTEC-IIM.	0.6441 (5)	0.5017 (3)	0.3850 (15)	0.4055 (10)
UWB best	0.6403	0.4571	0.4180	0.4080
RDST*	0.6327 (6)	0.4868 (5)	0.4388 (8)	0.4352 (6)
ELiRF-UPV	0.6327 (6)	0.4123 (12)	0.4404 (7)	0.4211 (7)
UWB submitted	0.6263 (7)	0.4404 (8)	0.4059 (12)	0.3902 (13)

* Random Decision Syntax Trees

Table 4: CodaLab results for Subtask B.

Feature	Subtask A submitted				Subtask A best			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
ALL*	0.6875	0.5988	0.6431	0.6202	0.7003	0.6195	0.6334	0.6264
ChN	-0.0434	-0.0586	+0.0482	-0.0137	-0.0446	-0.0636	+0.0225	-0.0246
BoM	+0.0000	+0.0000	+0.0000	+0.0000	-0.0038	-0.0044	-0.0064	-0.0054
BoT	-0.0051	-0.0037	-0.0193	-0.0110				
FW	+0.0000	+0.0006	-0.0032	-0.0012	-0.0064	-0.0089	-0.0032	-0.0061
LW	-0.0038	-0.0042	-0.0064	-0.0052				
List								
NSh	-0.0013	-0.0035	+0.0096	+0.0025				
POS	-0.0038	-0.0042	-0.0064	-0.0052				
POS-B					-0.0140	-0.0157	-0.0257	-0.0206
R-BoW	-0.0064	-0.0020	-0.0386	-0.0195	-0.0077	-0.0094	-0.0096	-0.0095
TF-IDF	+0.0000	+0.0000	+0.0000	+0.0000	-0.0064	-0.0068	-0.0129	-0.0098
V-BoW	+0.0051	+0.0066	+0.0032	+0.0050				
WN_1	+0.0013	+0.0030	-0.0064	-0.0014	-0.0089	-0.0107	-0.0129	-0.0117

* Original results achieved with all used features in the respective ablation study.

Table 5: Feature ablation study for Subtask A.

3.3 Subtask B

We classify tweets into one of four classes using the Maximum Entropy classifier and the features shown in Table 6. Blank space denotes that the corresponding feature has not been used.

4 Results and Experiments

Our results in Subtask A are in Table 3 and our results in Subtask B are in Table 4. The official evaluation metric was F1-score. The system settings and features were selected based on our pre-

Feature	Subtask B submitted				Subtask B best			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
ALL*	0.6263	0.4404	0.4059	0.3902	0.6403	0.4571	0.4180	0.4080
ChN	-0.0383	-0.0744	-0.0228	-0.0415	-0.0727	-0.0359	-0.0185	-0.0164
BoM	+0.0077	+0.0064	+0.0076	+0.0090	-0.0013	-0.0079	-0.0039	-0.0042
BoT								
FW	-0.0013	-0.0096	+0.0039	+0.0025	-0.0089	-0.0207	-0.0087	-0.0079
LW	+0.0089	+0.0011	+0.0091	+0.0094	+0.0000	-0.0068	+0.0000	-0.0001
List	+0.0051	+0.0001	+0.0065	+0.0066				
NSh	+0.0064	+0.0027	+0.0051	+0.0072	-0.0051	-0.0197	-0.0055	-0.0068
POS								
POS-B	+0.0051	+0.0034	+0.0069	+0.0089	-0.0102	-0.0289	-0.0145	-0.0176
R-BoW	+0.0064	+0.0062	+0.0055	+0.0097				
TF-IDF	+0.0000	-0.0042	+0.0024	+0.0035				
V-BoW	+0.0026	-0.0056	+0.0039	+0.0071				
WN_1	+0.0000	-0.0072	+0.0024	+0.0036				
$\text{WN}_{2,3}$	+0.0013	+0.0001	+0.0034	+0.0068				

* Original results achieved with all used features in the respective ablation study.

Table 6: Feature ablation study for Subtask B.

evaluation experiments using 10-fold cross validation on the training data for the team description *UWB submitted*. The team description *UWB best* represents the best settings according to the experiments on test data.

We performed ablation experiments to see which features are the most beneficial (see Table 5 and Table 6). Numbers represent the performance change when the given feature is removed.

We can see that many features in the submitted settings for both subtasks are not beneficial for the results, thus we remove them in the best settings for the given subtask. The best features apart from character n-grams include POS-B, FW, and BoM for both subtasks. In subtask A R-Bow, TF-IDF, and unigrams were also beneficial. In subtask B word shape n-grams were also helpful.

Detailed statistical analysis into the datasets and feature presence in the data would be needed in order to infer further insides.

5 Conclusion

We competed in both subtasks and ranked 4th in terms of accuracy in Subtask A and 7th in Subtask B. In terms of the F1-score measure we ranked 11th in Subtask A and 13th in Subtask B. However this comparison likely isn't fair because our system should not be be considered just constrained but strongly constrained.

Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports under the program NPU I and by university specific research project SGS-2016-018 Data and Software Engineering for Advanced Applications.

References

Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics.

Cristina Bosco, Viviana Patti, and Andrea Bolioli. 2013. Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE Intelligent Systems*, 28(2):55–63.

Tomáš Hercig, Tomáš Brychcín, Lukáš Svoboda, and Michal Konkol. 2016. UWB at SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 342–349. Association for Computational Linguistics.

Tomáš Hercig and Ladislav Lenc. 2017. The impact of figurative language on sentiment analysis. In

Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2017, Varna, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Michal Konkol. 2014. Brainy: A machine learning library. In Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi Zadeh, and Jacek Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499. Springer International Publishing.

Christine Liebrecht, Florian Kunneman, and Antal Van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 29–37, Atlanta, Georgia. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

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 Association for Computational Linguistics.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data Knowl. Eng.*, 74:1–12.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Paris, France. European Language Resources Association (ELRA).

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA. Association for Computational Linguistics.

Aline A Vanin, Larissa A Freitas, Renata Vieira, and Marco Bochernitsan. 2013. Some clues on irony detection in tweets. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 635–636. International World Wide Web Conferences Steering Committee.