EmotiKLUE at IEST 2018: Topic-Informed Classification of Implicit Emotions

Thomas Proisl, Philipp Heinrich, Besim Kabashi, Stefan Evert
Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl für Korpus- und Computerlinguistik
Bismarckstr. 6, 91054 Erlangen, Germany
{thomas.proisl,philipp.heinrich,besim.kabashi,stefan.evert}@fau.de

Abstract

EmotiKLUE is a submission to the Implicit Emotion Shared Task. It is a deep learning system that combines independent representations of the left and right contexts of the emotion word with the topic distribution of an LDA topic model. EmotiKLUE achieves a macro average F_1 score of 67.13%, significantly outperforming the baseline produced by a simple ML classifier. Further enhancements after the evaluation period lead to an improved F_1 score of 68.10%.

1 Introduction

The aim of the Implicit Emotion Shared Task (IEST; Klinger et al., 2018) is to infer emotion from the context of emotion words. The working definition of emotion for the shared task implies that emotion is triggered by the interpretation of a stimulus event (Scherer, 2005, 697), i. e. the cause of the emotion. Consequently, the data for the shared task have been compiled with the aim of including a description of the cause of the emotion. This has been accomplished by using distant supervision: The organizers collected tweets that contain exactly one of 21 emotion words belonging to six emotions (anger, fear, disgust, joy, sadness, surprise), where the emotion word has to be followed by *that*, *because* or *when* as likely indicators for a description of the cause of the emotion. The corpus collected this way comprises more than 190.000 tweets and is split into three data sets: 80% training, 5% trial and 15% test. The emotion words in the tweets are masked and participants of the shared task have to predict the emotion of the masked emotion word from its context.

EmotiKLUE, our submission to the shared task, is a deep learning system that learns independent representations of the left and right contexts of the emotion word, similar to Saeidi et al. (2016), who use n-gram representations for both the right and the left context around triggerwords in aspect-based opinion mining. Our intuition is that the distribution of the emotions is dependent on the topics of the tweets, therefore we train a Twitter-specific LDA topic model and explore different ways of combining the topic distributions with the left and right contexts in order to predict the emotions. EmotiKLUE is available on GitHub.[1]

2 Related Work

Emotion detection has been an important topic in natural language processing, particularly in the subfield of opinion mining, for several years. The shallowest approaches deal with sentiment polarity detection, either classifying utterances into categories ranging from *negative* via *neutral* to *positive*, or regressing towards a score typically ranging from -1 to 1 (see, for example, Proisl et al., 2013; Evert et al., 2014). Further tasks involve the automatic computation of stances (*in favor of* vs. *against*) towards pre-specified topics (Mohammad et al., 2017). Predicting more sophisticated categories of emotion than in the task at hand has been a more recent phenomenon. Generally, the approaches can be classified into two groups, namely rule-based approaches on the one hand and the far more common machine learning approaches on the other.

We give a short list of related work here, for a more comprehensive listing see the task description (Klinger et al., 2018). A survey of emotion detection from text and speech is given by Sailunaz et al. (2018). For a linguistic analysis of implicit emotions see Lee (2015). An approach to implicit emotion detection based on textual inference is presented by Ren et al. (2017).

[1] https://github.com/tsproisl/EmotiKLUE

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 235–242
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

As an example for rule-based emotion detection we mention Udochukwu and He (2015), who use a pipeline approach based on the OCC-Model (Ortony et al., 1988), without emotion-bearing words.

More recent work deals with ML and deep learning approaches. Rout et al. (2018) use both unsupervised and supervised approaches with different machine learning algorithms such as multinomial naive bayes, maximum entropy, and support vector machines on unigram feature matrices and report F_1-scores of above 99% when disambiguating tweets according to seven emotion categories. However, since their text data are selected via a keyword-filter containing exactly the words representing the emotion which in turn can be used as features by the machine learner at hand, their high accuracy values are unsurprising.

Other tasks, such as detecting the emotion *stimulus* in emotion-bearing sentences are more challenging; Ghazi et al. (2015) e. g. use a conditional random fields classifier and report F_1-scores of up to 60% for finding the stimulus in their self-constructed data set. Finally, Firdaus et al. (2018) use different latent features such as emotion and sentiment as input to predict user behaviour (e. g. the act of *retweeting*).

3 System Description

3.1 Data Preprocessing and Additional Data

The data sets released by the organizers of the shared task contain the full text of the tweets, with the emotion word, usernames and URLs being substituted by placeholders. We tokenize the text with the web and social media tokenizer SoMaJo[2] (Proisl and Uhrig, 2016) and convert it to lowercase.

In addition to the official data sets, we use two resources: ENCOW14[3] (Schäfer and Bildhauer, 2012; Schäfer, 2015) and an in-house collection of 114 million deduplicated English tweets (see Schäfer et al. (2017) for the deduplication algorithm), collected between February 2017 and June 2018.[4] We tokenize the tweets with SoMaJo (but not ENCOW14, which is already tokenized), mask usernames and URLs and convert the text to lowercase.

3.2 Representations derived through unsupervised methods

We use our in-house collection of tweets to create Twitter-specific word embeddings and topic models.

Using the Gensim[5] (Řehůřek and Sojka, 2010) implementation of word2vec (Mikolov et al., 2013a,b), we create four sets of embeddings for all words with a minimum frequency of 5: 100- and 300-dimensional vectors using the skip-gram approach and 100- and 300-dimensional vectors using the CBOW approach.

Our intuition is that the distribution of the emotion words depends on the topics of the tweets. To capture these topics, we use Gensim and create an LDA topic model (Blei et al., 2003) with 100 topics based on the most recent 10 million tweets in our collection (ignoring words that only occur once).

3.3 Additional Data for Pretraining

We compile an additional data set from ENCOW14 and our collection of tweets that we use to pretrain our model. To this end, we select tweets and ENCOW14 sentences with a maximum length of 110 words that contain a single emotion word from the following set of emotion words: *afraid*, *angry*, *disgusted*, *disgusting*, *happy*, *sad*, *surprised*, *surprising*. This list of emotion words was determined by a cursory glance at the official training data and happens to be a subset of the 21 emotion words used by the task organizers (which were only revealed after the evaluation period). Note that we do not restrict the contexts in which the emotion words occur, i. e. the emotion words do not have to be followed by *that*, *because* or *when*. After balancing the data, we have approximately 159.000 items per class.

3.4 Network Architecture

We experiment with three variants of a neural network architecture implemented using Keras[6] (Chollet et al., 2015) and visualized in Figure 1.

The word-level representations for the left and right contexts of the emotion word that are returned by the embedding layers are fed into

[2] https://github.com/tsproisl/SoMaJo

[3] http://corporafromtheweb.org/encow14

[4] The overlap of the released data sets with our in-house collection of tweets is negligible. Our collection contains less than 0.6% of the tweets from the released data sets: 775 from the training set (0.51%), 49 from the trial set (0.51%) and 163 from the test set (0.57%).

[5] https://radimrehurek.com/gensim

[6] https://keras.io

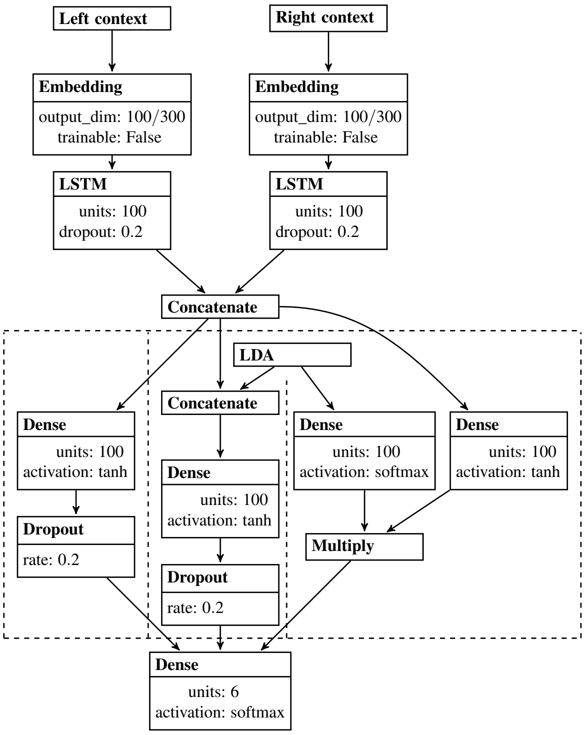

Figure 1: Architecture of the three model variants

two unidirectional LSTM layers (Hochreiter and Schmidhuber, 1997; Gers et al., 2000): A left-to-right layer for the left context from the beginning of the tweet to the masked emotion word, and a right-to-left layer for the right context from the end of the tweet to the masked emotion word. The hidden states of the two LSTM layers are concatenated. Now, we explore three variants of incorporating the 100-dimensional LDA topic distribution into the model:

1. We do not use LDA topics. The output of the LSTMs is fed to a dense layer, followed by a dropout layer and finally a softmax output layer.

2. We use LDA topics as features alongside the LSTM output. The LDA topic distribution and the output of the LSTMs are concatenated. The result is fed to a dense layer, followed by a dropout layer and finally a softmax output layer.

3. We use LDA topics as filter. The output of the LSTMs is fed to a dense layer to reduce dimensionality. The LDA topic distribution is fed to a softmax layer. The output of the two layers is combined using element-wise multiplication. The result is fed to the final softmax output layer.

model	trial	test
train-skip100-nolda	64.06	65.14
train-skip100-ldafeat	64.46	65.10
train-skip100-ldafilt	64.56	65.03
train-skip300-nolda	65.93	66.33
train-skip300-ldafeat	66.05	66.35
train-skip300-ldafilt	65.18	65.79
add-skip100-nolda	52.01	52.12
add-skip100-ldafeat	52.49	52.84
add-skip100-ldafilt	51.29	51.88
add-skip300-nolda	55.28	55.49
add-skip300-ldafeat	55.22	55.11
add-skip300-ldafilt	52.76	52.68
add+train-skip100-nolda	65.19	66.55
add+train-skip100-ldafeat	65.71	66.02
add+train-skip100-ldafilt	65.67	65.94
add+train-skip300-nolda	67.05	**67.50**
add+train-skip300-ldafeat	**67.17**	67.08
add+train-skip300-ldafilt	66.43	67.00
add+train+trial-skip300-ldafeat (subm.)		67.13

Table 1: Results for models using skip-gram-based embeddings (macro F_1)

We train each model for a maximum of 20 epochs with a batch size of 160, using the Adam optimizer (Kingma and Ba, 2014) to minimize categorical crossentropy. If the validation loss (determined on the trial data) fails to improve for two consecutive epochs, training stops early.

4 Results and Error Analysis

4.1 Experiments

We have three different network architectures that differ in the way they use LDA topic distributions. We have four sets of embeddings that differ in size and training objective. And we have three options for the training data (only the official training data, only our additional data, or training on the latter and retraining on the former). In order to quantify the impact of the individual choices, we train and evaluate all 36 possible models. Results for models using skip-gram-based embeddings are shown in Table 1 and results for models using CBOW-based embeddings in Table 2. The evaluation metric used is the macro average of the F_1 scores of the six classes.

The exact numbers listed in Tables 1 and 2 should not be taken too seriously as they are subject to some small amout of random variation due

model	trial	test
train-cbow100-nolda	63.75	64.07
train-cbow100-ldafeat	62.81	63.20
train-cbow100-ldafilt	63.39	63.24
train-cbow300-nolda	64.09	63.91
train-cbow300-ldafeat	64.00	63.94
train-cbow300-ldafilt	63.61	63.49
add-cbow100-nolda	49.64	50.14
add-cbow100-ldafeat	48.16	48.55
add-cbow100-ldafilt	48.69	48.81
add-cbow300-nolda	51.26	50.70
add-cbow300-ldafeat	51.25	51.48
add-cbow300-ldafilt	49.31	49.01
add+train-cbow100-nolda	63.10	64.03
add+train-cbow100-ldafeat	64.46	64.08
add+train-cbow100-ldafilt	63.42	63.60
add+train-cbow300-nolda	64.34	64.74
add+train-cbow300-ldafeat	64.26	64.66
add+train-cbow300-ldafilt	63.83	63.64

Table 2: Results for models using CBOW-based word embeddings (macro F_1)

to differences in the initialization of the weights and the shuffling of the training data.[7] However, since all the individual options have been used at least nine times, we can still make some fairly reliable claims about their usefulness.

The most obvious observation is that the official training data lead to much better results than our additional data (+12.97 on average). This is probably due to two reasons: We only use a subset of the emotion words that have been used in the official data sets and, more importantly, we use all instances of the emotion words and not only those that are followed by something that is likely to be a description of the cause of the emotion. However, first training the model on the additional data and then retraining it on the official training data is benefitial (+1.96).

We can also see that word embeddings based on the skip-gram approach consistently outperform those based on the CBOW approach (+2.55). 300-dimensional embeddings are notably better than 100-dimensional embeddings (+1.19), an effect that is more pronounced for the skip-gram-based embeddings (+1.57) than for the CBOW-based ones (+0.80).

The LDA topic distributions only have a positive effect when used as additional features alongside the LSTM output – and even then the effect is small and only positive for models using skip-gram-based embeddings (+0.08) and negative for models using CBOW-based embeddings (−0.24). Using the LDA topic distribution as a filter usually has a negative effect (−0.76).

Consequently, for our submission to the shared task, we chose the second network architecture (LDA topic distribution as feature), used 300-dimensional skip-gram embeddings and trained the model first on our additional data and retrained it on the official training and trial data. That model achieved a macro average F_1 score of 67.13 on the test data and took the tenth place in the shared task. For comparison, Klinger et al. (2018) report that human performance on this task is approximately 45%, the MaxEnt uni- and bigram classifier used as a baseline system achieved 59.88% and the best submission (Rozental et al., 2018) 71.45%.

4.2 Error Analysis

We present detailed error analyses in Table 3 in form of an extensive confusion matrix including label confusion per triggerword in the test data. We downloaded all available tweets used in the shared task via the Twitter API[8] to gain access to the actual triggerwords. For reasons of interpretability we report absolute marginal frequencies and relative frequencies of predicted label per real label and triggerword.[9] This corresponds to recall (true-positive-rate) for those cases where the prediction equals the true label and false-negative-rate (FNR) per class for all other cases.

Recall is rather similar across labels: The highest rate can be achieved for *joy* (78%), the lowest is achieved for *sad* (59%). High FNRs have to be reported for confusing *anger*, *disgust*, and *fear* with *surprise* (11% and 10%), as well as *sad* with *anger* and *disgust* (each 11%).

Looking at the recall values per triggerword, explanations for the macro-values are not far to seek:

1. Performance is generally higher for those triggerwords that have been manually se-

[7] The 95%-confidence interval for the performance of the add+train-skip300-ldafeat model on the test data is 67.12 ± 0.34, for example (estimated from 20 instances of the model).

[8] https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-lookup

[9] The difference in absolute numbers between label-based confusions and triggerword-based confusions are due to the fact that not all tweets can be retrieved from the API – once a tweet is e. g. deleted by a user, it is no longer accessible for others either.

	anger	**disgust**	**fear**	**joy**	**sad**	**surprise**	**total**
anger	**0.61**	0.09	0.08	0.06	0.06	0.11	**4794**
angry	***0.62***	*0.07*	*0.08*	*0.07*	*0.06*	*0.09*	*2893*
furious	***0.57***	*0.11*	*0.06*	*0.04*	*0.05*	*0.18*	*1292*
disgust	0.08	**0.67**	0.04	0.03	0.07	0.11	**4794**
disgusted	*0.14*	***0.53***	*0.06*	*0.05*	*0.03*	*0.19*	*2065*
disgusting	*0.03*	***0.79***	*0.01*	*0.01*	*0.10*	*0.05*	*2398*
fear	0.08	0.04	**0.69**	0.05	0.04	0.10	**4791**
afraid	*0.05*	*0.02*	***0.76***	*0.04*	*0.03*	*0.08*	*1693*
fearful	*0.10*	*0.03*	***0.69***	*0.05*	*0.03*	*0.10*	*315*
frightened	*0.11*	*0.11*	***0.49***	*0.05*	*0.04*	*0.20*	*324*
scared	*0.10*	*0.05*	***0.62***	*0.06*	*0.05*	*0.12*	*1648*
joy	0.06	0.02	0.04	**0.78**	0.04	0.06	**5246**
cheerful	*0.09*	*0.05*	*0.05*	***0.64***	*0.07*	*0.09*	*56*
happy	*0.06*	*0.02*	*0.04*	***0.79***	*0.04*	*0.06*	*4215*
joyful	*0.05*	*0.04*	*0.08*	***0.61***	*0.09*	*0.12*	*97*
sad	0.11	0.11	0.06	0.07	**0.59**	0.06	**4340**
depressed	*0.21*	*0.08*	*0.09*	*0.10*	***0.46***	*0.06*	*642*
sad	*0.09*	*0.12*	*0.05*	*0.06*	***0.62***	*0.06*	*2751*
sorrowful	*0.00*	*0.12*	*0.00*	*0.50*	***0.25***	*0.12*	*8*
surprise	0.08	0.09	0.07	0.05	0.03	**0.68**	**4792**
astonished	*0.08*	*0.13*	*0.07*	*0.04*	*0.01*	***0.66***	*350*
astounded	*0.07*	*0.17*	*0.09*	*0.03*	*0.01*	***0.63***	*263*
shocked	*0.12*	*0.06*	*0.08*	*0.06*	*0.03*	***0.65***	*1021*
startled	*0.10*	*0.06*	*0.22*	*0.04*	*0.01*	***0.57***	*228*
stunned	*0.12*	*0.10*	*0.08*	*0.07*	*0.01*	***0.62***	*500*
surprised	*0.07*	*0.05*	*0.06*	*0.06*	*0.01*	***0.74***	*1223*
surprising	*0.02*	*0.11*	*0.01*	*0.01*	*0.12*	***0.74***	*805*
total	**4841**	**4801**	**4633**	**5305**	**3732**	**5445**	**28757**

Table 3: Confusion Matrix for the six predicted emotion categories (columns) for each real emotion and each triggerword (rows) in the test data

lected by us for producing additional training data (see Section 3.3): *angry* (62%) shows higher recall than *furious* (57%), *afraid* (76%) and *happy* (79%) perform best in the *fear* and *joy* categories, respectively, and *surprised* and *surprising* (each 74%) are the best predictors for *surprise*.

2. Rare triggerwords generally lead to worse results. The most obvious example is *sorrowful*, which we only observed 28 times in the training data (8 times in the test data) and which yields 25% recall for predicting category *sad*, confusing it in half of the cases with *joy*. Additionally, *cheerful* and *joyful* (361 and 536 observations in the training data, respectively) perform lower than *happy* (22348 observations) – although admittedly *happy* had already been pre-selected for additional training as mentioned above.

3. Many confusions can also be explained from a psycho-linguistic point of view when looking at the actual corpus. Instances involving the triggerword *disgusted* e. g. are frequently categorized as *anger* by our system. Corpus evidence shows that these words are hard to disambiguate:

   - Hindu women should be [#TRIGGERWORD#] when Law Panel says Father-In-Law should pay alimony, what next

model	trial	test
add2-skip300-ldafeat	56.66	56.98
add2+train-skip300-ldafeat	67.34	67.47
300-train-skip300-ldafeat	66.14	66.68
300-add-skip300-ldafeat	57.10	57.29
300-add+train-skip300-ldafeat	67.89	68.06
300-add2-skip300-ldafeat	58.35	58.49
300-add2+train-skip300-ldafeat	67.98	68.10

Table 4: Results for the post-analysis experiments (macro F_1)

women are property of Father-In-Law?

- I wake up [#TRIGGERWORD#] because I know you doin me wrong but u dont think its nothing wrong with being in a verbal relationship with another gal

It is hard to see how one could reliably predict the "real" emotion (*disgust*) in the above examples, since *anger* – as predicted by our system – seems to be an equally sensible guess. Similar instances can be found for other confusions, most notably when predicting *anger* in case of the triggerword *depressed*.

4.3 Post-analysis experiments

The analysis in the previous section has shown that our system performs better on the more frequent words that we used for compiling our additional data than on the less frequent words. Therefore, we recompile our additional data as described in Section 3.3 but for all of the 21 emotion words that occur in the official data. After balancing the data, this results in approximately 163.000 items per class.

We take the model versions from Section 4.1 that are the basis for our submission and replace the additional data with the updated version. The new models (prefixed with "add2" in Table 4) improve on the old ones both when using only the additional data (+1.66) and when retraining on the official training data (+0.28).

It is also worth pointing out that so far we have not fine-tuned the hyperparameters of our model. As a first step in that direction, we try to use more units in the hidden layers and increase the size of all hidden layers to 300 units (models prefixed with "300-add" in Table 4). This boosts the performance both when using only the additional data (+2.03) and when retraining on the official training data (+0.85).

Combining the recompiled additional data and the larger hidden layers yields further improvements (models prefixed with "300-add2" in Table 4). The retrained model is approximately 1 point better than our submission and would have taken the eighth place in the shared task.

A further error analysis shows that the additional training data indeed yield the desired effect: Recall for category *angry* improves from 61% to 66%, largely due to better recall in the case of the triggerword *furious* (rising from 57% to 65%). Further improvements can be found in almost all categories, namely for *fear* (69% to 72%, especially *frightened*: 49% to 53%), *joy* (78% to 79%, with recall for *joyful* rising from 61% to 65% and for *cheerful* from 64% to 70%), and *sad* (59% to 62%, triggerword *depressed* up two points from 46% to 48%). However, the additional training data had an adverse effect on category *surprise*; here recall falls from 68% to 65%, with almost all triggerwords dropping a couple of points, the worst being *surprised*, falling from 74% to 69%.

Finally, we want to take a closer look at the contribution of the LDA topic distribution. To this end, we have trained 20 instances of the 300-add2+train-skip300-ldafeat and 300-add2+train-skip300-nolda models and have calculated the means and 95%-confidence intervals. As it turns out, both model variants perform identically on the trial data. On the test data, there are some minor differences but the performance means lie within one standard deviation of each other. This means that our choice of concatenating the LDA topic distribution of the tweet to the LSTM does not have a statistically significant result..

5 Conclusion

We presented EmotiKLUE, a topic-informed deep learning system for detecting implicit emotion. Our experiments showed that for this task skip-gram-based word embeddings outperform CBOW-based embeddings. Additional data, that – on their own – yield rather poor results, improve the performance when used for pretraining the model. LDA topic models, that we initially believed to have a small positive effect, turned out to not contribute significantly.

The error analysis shows that the objective as set in the shared task at hand is rather difficult: With many instances of tweets showing prima facie ambiguous emotions, it is unsurprising that even perfectly trained classifiers will not be able to achieve 100% accuracy when using the textual data alone.

Future work could nonetheless involve more experimentation with the hyperparameters of the network, e. g. number, size and activation of the hidden layers, choice of regularization strategy and optimizer, etc.

The software is available on GitHub.[10]

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

François Chollet et al. 2015. Keras. `https://keras.io`.

Stefan Evert, Thomas Proisl, Paul Greiner, and Besim Kabashi. 2014. SentiKLUE: Updating a polarity classifier in 48 hours. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 551–555, Dublin. Association for Computational Linguistics.

Syeda Nadia Firdaus, Chen Ding, and Alireza Sadeghian. 2018. Topic specific emotion detection for retweet prediction. *International Journal of Machine Learning and Cybernetics*, pages 197–203.

Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *Computational Linguistics and Intelligent Text Processing. CICLing 2015*, pages 152–165.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Roman Klinger, Orphée de Clercq, Saif M. Mohammad, and Alexandra Balahur. 2018. IEST: WASSA-2018 Implicit Emotions Shared Task. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Brussels. ACL.

Sophia Yat Mei Lee. 2015. A linguistic analysis of implicit emotions. In *Chinese Lexical Semantics - 16th Workshop, CLSW 2015, Beijing, China, May 9-11, 2015, Revised Selected Papers*, pages 185–194.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and sentiment in tweets. *Special Section of the ACM Transactions on Internet Technology on Argumentation in Social Media*, 17(3).

A. Ortony, G. Clore, and A. Collins. 1988. *Cognitive Structure of Emotions*. Cambridge University Press.

Thomas Proisl, Paul Greiner, Stefan Evert, and Besim Kabashi. 2013. KLUE: Simple and robust methods for polarity classification. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, pages 395–401, Atlanta, GA. Association for Computational Linguistics.

Thomas Proisl and Peter Uhrig. 2016. SoMaJo: State-of-the-art tokenization for German web and social media texts. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 57–62, Berlin. ACL.

Han Ren, Yafeng Ren, Xia Li, Wenhe Feng, and Maofu Liu. 2017. Natural logic inference for emotion detection. In *Proceedings of CCL 2017 and NLP-NABD 2017*, pages 424–436.

Jitendra Kumar Rout, Kim-Kwang Raymond Choo, Amiya Kumar Dash, Sambit Bakshi, Sanjay Kumar Jena, and Karen L. Williams. 2018. A model for sentiment and emotion analysis of unstructured social media text. *Electronic Commerce Research*, 18(1):181–199.

Alon Rozental, Daniel Fleischer, and Zohar Kelrich. 2018. Amobee at IEST 2018: Transfer learning from language models. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Brussels. ACL.

Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. 2016. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1546–1556, Osaka, Japan. The COLING 2016 Organizing Committee.

[10] `https://github.com/tsproisl/EmotiKLUE`

Kashfia Sailunaz, Manmeet Dhaliwal, Jon Rokne, and Reda Alhajj. 2018. Emotion detection from text and speech: a survey. *Social Network Analysis and Mining*, 8(1):28:1–28:26.

Klaus R. Scherer. 2005. What are emotions? And how can they be measured? *Social Science Information*, 44(4):695–729.

Fabian Schäfer, Stefan Evert, and Philipp Heinrich. 2017. Japan's 2014 General Election: Political Bots, Right-Wing Internet Activism and PM Abe Shinzō's Hidden Nationalist Agenda. *Big Data*, 5(4):294–309.

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, pages 28–34, Lancaster. UCREL, IDS.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 486–493, Istanbul. ELRA.

Orizu Udochukwu and Yulan He. 2015. A rule-based approach to implicit emotion detection in text. In *Proceedings of NLDB 2015*, pages 197–203.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 46–50, Valletta. ELRA.