SymantoResearch at SemEval-2019 Task 3: Combined Neural Models for Emotion Classification in Human-Chatbot Conversations

Angelo Basile*, Marc Franco-Salvador*, Neha Pawar*, Sanja Štajner*,
Mara Chinea Rios, and Yassine Benajiba

Symanto Research, Nürnberg, Germany
{angelo.basile, marc.franco, neha.pawar, sanja.stajner,
mara.chinea, yassine.benajiba}@symanto.net

Abstract

In this paper, we present our participation to the EmoContext shared task on detecting emotions in English textual conversations between a human and a chatbot. We propose four neural systems and combine them to further improve the results. We show that our neural ensemble systems can successfully distinguish three emotions (SAD, HAPPY, and ANGRY), and separate them from the rest (OTHERS) in a highly-imbalanced scenario. Our best system achieved a 0.77 F_1-score and was ranked fourth out of 165 submissions.

1 Introduction

Detecting emotions in text is a key task in many scenarios, such as social listening, personalised marketing, customer caring, or in building emotionally intelligent chat-bots: in this last case, the task complexity increases, since a bot's response might influence the user's emotion.

The EmoContext shared task (Chatterjee et al., 2019) was posed as a sequence classification task. Given a set of three conversational turns (human–bot–human), the goal is to predict the emotion of the third turn. The label space contains the emotions SAD, ANGRY and HAPPY, and the label OTHERS denoting anything else (emotional or non-emotional), as illustrated in Table 1.

In this paper, we present our approaches to EmoContext shared task, and describe our best system in details. Additionally, we show that: (a) this task is very difficult even for humans (Section 2.2); (b) for this task, neural approaches outperform a strong non-neural baseline (Section 4); (c) an ensemble of neural systems with different architectures significantly outperforms the best neural model in isolation (Section 4).

2 Data

The data released by the organisers consist of English user-chatbot interactions occurring in an Indian chat room. An overview of the dataset is provided in Table 2. It can be seen that the label distribution is highly imbalanced, and different for the training set than for the development (dev) and test sets (a 14:18:18:50 distribution for the training set, and a 5:5:5:85 distribution for the dev and test sets). To overcome this issue we tested three strategies: (1) down-sampling the dataset to its smallest class; (2) up-sampling the emotion-related labels with an in-house dataset; and (3) up-sampling by duplicating a random portion of the dataset. None of these solutions worked, and therefore, we trained our best models using the data provided by the organisers.

2.1 Preprocessing

The language of this corpus presents many of the features of micro-blogging language: large use of contractions (e.g. *I'm gonna bother*), elongations (e.g. *a vacation tooooooo!*), non-standard use of punctuation (e.g. *gonna explain you later..!*), incorrect spelling (e.g. *U r*).

To properly handle this language, we build a simple preprocessing pipeline which consists of: (1) the NLTK `TweetTokenizer` (Bird and Loper, 2004); and (2) a normalisation strategy that reduces sparseness by lowercasing all the words and converting elongations like *looool* to *lol*. These steps are used in all the experiments. Some of our models use additional preprocessing described further in the text.

* The first four authors have contributed equally to this work and are ordered alphabetically.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 330–334
Minneapolis, Minnesota, USA, June 6–7, 2019.

ID	TURN 1	TURN 2	TURN 3	LABEL
71	Not good	:(why not..?	Been sick for one week	SAD
78	I hate Siri and it's friends	if you hate them , they are not your friends then xD	Yeah and u r Siri's friend so I hate utoo	ANGRY
91	Now I'm doing my dinner	I can see you!	How can you see me??	OTHERS
140	How about you tired of life or just your day?	Aha I' happy today, thanks for asking	Wow great..!	HAPPY

Table 1: Examples from the training dataset.

CLASS	TRAIN	DEV	TEST
SAD	5463	125	250
HAPPY	4243	142	284
ANGRY	5506	150	298
OTHERS	14948	2338	4677
total	30160	2755	5509

Table 2: Distribution of classes.

2.2 Manual Validation

To check how difficult this task is, for a trained human annotator, and get an estimate of the expected upper limit for our classification models, we asked two fluent (but non-native) English speakers with previous annotation experience to label 300 randomly selected instances from the dev set. The annotators achieved the official F_1-score of 0.73 and 0.72 against the 'gold' labels, and a 0.71 F_1-score among themselves. The only observed misclassifications between "emotional" classes were those between SAD and ANGRY. The highest number of disagreements the annotators had was between the OTHERS and the "emotional" classes. This showed that: (1) the task is naturally difficult (the trained human annotators reach 0.73 F_1-score at the most); (2) the main problem is distinguishing between the OTHERS class and the "emotional" classes.

3 Experimental Setup

We first randomly selected two times 2754 instances from the official training set, maintaining the class ratio that was announced for the official dev and test sets (4:4:4:88) resulting in 110 instances for the SAD, ANGRY and HAPPY, and 2424 instances for the OTHERS class. These two datasets we refer to as intDev and intTest sets, while the rest of the training dataset we refer to as intTrain.

We train and tune our four neural models (Section 3.1) using intTrain and intDev sets, and test them on the intTest, and the official dev and test sets (in different phases of the competition). We further experiment with combining their softmax output per class probabilities (Section 3.2).

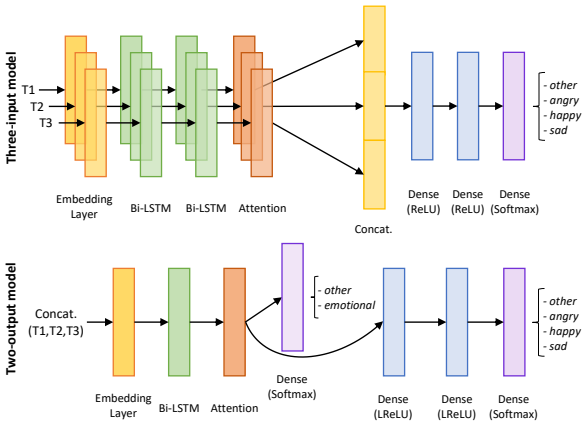

Figure 1: Model architectures for the three-input (IN3) and two-output (OUT2) models.

As a strong non-neural baseline we set up a linear SVM model with word and character n-grams (1-6) as features.[1]

3.1 Neural Models

We propose four neural network models that slightly differ on their objective.

3.1.1 Three-Input Model (IN3)

Having the three conversation turns (T1, T2, and T3), we explicitly represent the position of each sequence in the conversation by creating an input branch for each turn. The branches are identical and represent the text using word embeddings that feed a 2-layer bidirectional Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). An attention mechanism (Yang et al., 2016) combines its hidden states. This architecture allows to independently process and attend to the most relevant parts of T1, T2, and T3. The information is later combined by a simple concatenation and few fully connected dense layers. The model architecture is shown in Figure 1.

We use a proprietary model © Symanto Research to obtain 300-dimensional word embeddings on the English Wikipedia. The performance

[1]For the implementation of the baseline we use `scikit-learn` (Pedregos et al., 2011). All the neural models are based on `Tensorflow` (Abadi et al., 2016), and for the ensemble models we use `Weka` (Hall et al., 2009).

of this representation is comparable with fastText (Bojanowski et al., 2017) but the resulting embedding model is fifty times lighter. We apply 10% dropout on the output of the embedding and concatenation layers, and layer normalisation (Ba et al., 2016) after the concatenation and before the output softmax.

3.1.2 Two-Output Model (OUT2)

Motivated by the findings of the manual validation (Section 2.2), we build this model in an attempt to ease the *emotional* vs. OTHERS classification. For this reason, we use a multi-task learning approach and add an auxiliary output whose label space conflated the ANGRY, HAPPY, and SAD labels into a single *emotional* one. We hypothesise that this approach is well suited to our unbalanced scenario, with the dominant OTHERS class.

The model architecture is similar to our three-input one (see Figure 1). However, the three conversational turns (T1, T2, and T3) are fed to the model as a single concatenated input, with additional tokens to mark the turn boundaries. The auxiliary output is connected to the output of the attention. This forces the attention weights to favour the *emotional* vs. OTHERS task.

We use the pretrained word embeddings described in Section 3.1.1. Our dense layers use the leaky version (LReLU) of the Rectified Linear Unit (ReLU) activation. In addition, we use the attention mechanism (He et al., 2017). Finally, we use the batch normalisation (Ioffe and Szegedy, 2015) to process the attention output.

3.1.3 Sentence-Encoder Model (USE)

As an exploration in transfer-learning, we build a simple feed-forward network together with a fine-tuned Universal Sentence Encoder (Cer et al., 2018). As input, we use the first (T1) and the last (T3) turn of the conversation, as we observed that adding the second turn (T2) leads to lower performances of this model.

3.1.4 BERT Model (BERT)

We fine-tune a BERT-base model (Devlin et al., 2018), modelling the problem as a sentence-pair classification problem: we use the first and the third conversational turn (T1 and T3) as the first and the second sentence respectively, completely ignoring the utterance by the bot (T2). We use this model in combination with a lexical normalisation system (van der Goot and van Noord, 2017).

We also built a neural model combining BERT, IN3, and OUT2, but it resulted in lower performance than any of those models separately, and is thus not presented here.

3.2 Ensemble Models

As we noticed that our neural systems have different strengths and weaknesses on the "emotional" classes (see Table 4), we combine them by using the softmax output probabilities of each class from all four models (16 features in total) and training several classification algorithms: Naïve Bayes (John and Langley, 1995), Logistic Regression (le Cessie and van Houwelingen, 1992), Support Vector Machines (Keerthi et al., 2001) with normalization (SVM-n) or standardization (SVM-s), JRip rule learner (Cohen, 1995), J48 (Quinlan, 1993), Random Forest (Breiman, 2001), and various meta-learners on top of them or their subsets.

The neural systems are trained and tuned on the intTrain and intDev sets, and their per class probabilities are obtained for the intTest, dev, and test sets. The ensemble models are then trained on the intTest+dev set and tested on the official test set. For this second classification stage, we thus have 5509 instances for training (intTest+dev) and 5509 for testing (the official test set).

4 Results

We evaluate our systems using precision (P) and recall (R) per each emotional class, and the micro F_1-score over the three "emotional" classes (the metric used by the task organisers for the official evaluation). The results for the baseline and the four neural systems are presented in Table 4. The results of the best ensemble models (trained on the per class probabilities of the four neural models) are presented in Table 5. We can notice that:

(1) Our best neural system (IN3) reaches .73 on the intTest set and .72 on the official test set.

(2) All our neural systems have a noticeably higher recall on the HAPPY and SAD classes on the intTest set than on the official dev and test sets.

(3) Our two best neural systems (IN3 and OUT2) have a noticeably lower precision on the HAPPY and SAD classes on the intTest set than on the official dev and test sets.

(4) Ensemble models reach .77 for three classification algorithms in the 10-fold cross-validation setup on the intTest+dev set, and that score is maintained on the official test set only by SVM.

ID	TURN 1	TURN 2	TURN 3	GOLD	OUR
388	Ok... No problem	ok i hope what you stay ok, be safe:)	Fuck off	OTHERS	ANGRY
4035	Am so pissed	one had been there for A MONTH	I'm so pissed	SAD	ANGRY
4129	yes. tomorrow :D	Yay, you.	hehehe gives sly smirk	OTHERS	HAPPY
397	What madness r u speaking abt?	what language do u think I'm speaking	English??	HAPPY	OTHERS
1640	You don't have it from outside	You have form the inside? Are you a sock?	??	ANGRY	OTHERS
253	wt u mean	I mean rest if the year	????	SAD	OTHERS

Table 3: Error analysis on the official test set.

SYSTEM	TEST	SAD		ANGRY		HAPPY		F
		P	R	P	R	P	R	
OUT2	int	.66	.90	.67	.87	.55	.84	.73
	dev	.75	.78	.65	.78	.62	.77	.72
	test	.71	.82	.64	.76	.65	.74	.71
IN3	int	.68	.92	.68	.84	.53	.88	.73
	dev	.71	.78	.67.	.81	.61	.76	.72
	test	.70	.78	.67	.81	.62	.76	.72
USE	int	.65	.84	.44	.93	.50	.89	.65
	dev	.68	.78	.47	.92	.56	.78	.66
	test	.68	.76	.48	.92	.58	.76	.66
BERT	int	.59	.92	.48	.92	.47	.88	.65
	dev	.61	.82	.51	.91	.50	.70	.64
	test	.59	.82	.54	.89	.51	.74	.65
baseline	int	.51	.89	.47	.85	.46	.81	.60
	dev	.57	.78	.47	.86	.54	.77	.63
	test	.55	.82	.48	.90	.52	.70	.63

Table 4: Results of our four neural systems and the strong non-neural baseline on the intTest (int), and the official development (dev) and test (test) sets.

SYSTEM	TEST	SAD		ANGRY		HAPPY		F
		P	R	P	R	P	R	
Logistic	CV	.81	.81	.76	.80	.72	.73	.77
	test	.83	.76	.74	.77	.77	.64	.75
SVMn	CV	.81	.83	.74	.80	.66	.76	.77
	test	.82	.80	.73	.79	.75	.72	**.77**
RanForest	CV	.82	.82	.81	.73	.73	.67	.76
	test	.83	.76	.77	.72	.80	.63	.75

Table 5: Results of our best ensemble models in a 10-fold cross-validation setup on intTest+dev (CV), and training on intTest+dev and testing on test set. Our best system submitted to the competition is marked in bold.

5 Error Analysis

The confusion matrix for our best system is given in Figure 2. The highest number of confusions is between the HAPPY and OTHERS classes, followed by confusions between the ANGRY and OTHERS.

Given the findings of our manual validation (Section 2.2), we performed an additional experiment. All instances for which our best system did not predict the gold label (355 instances), we presented to one of our annotators together with its gold and predicted labels (in random order), and asked him to choose the correct one, or assign a NOT SURE label. The annotator chose the label predicted by our system in 46% of the cases, the gold label in 39% of the cases, and in 15% of the cases the annotator was not sure. Several examples of instances for which the predicted label did not match the "gold" label are presented in Table 3.

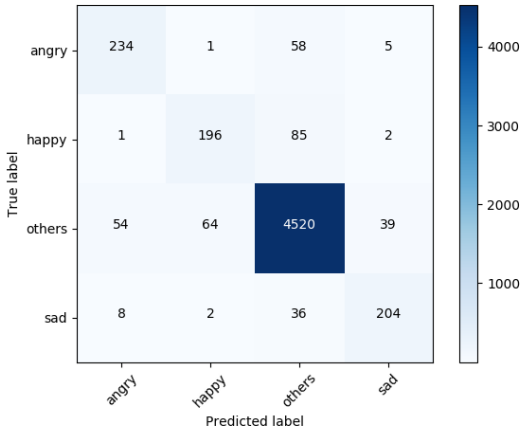

Figure 2: Confusion martix for the best model.

6 Conclusions

We presented our most successful approaches to the EmoContext shared task, with the goal of predicting the emotion (SAD, HAPPY, ANGRY, or OTHERS) in the third turn of a human–chatbot-human interaction, with an additional challenge of having a very unbalanced distribution of classes.

We showed that the task is difficult even for trained human annotators, and that our best neural systems can reach the human performance (.72 F-measure). Furthermore, we showed that a SVM classifier trained on the softmax output per class probabilities of four different neural systems can improve results scoring a .77 F_1-measure over the three emotional classes, and reaching thus the fourth place in the official competition.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Saskia le Cessie and Johannes C. van Houwelingen. 1992. Ridge Estimators in Logistic Regression. *Applied Statistics*, 41(1):191–201.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext: Contextual emotion detection in text. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019)*, Minneapolis, Minnesota.

William W. Cohen. 1995. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Rob van der Goot and Gertjan van Noord. 2017. Monoise: Modeling noise using a modular normalization system. *arXiv preprint arXiv:1710.03476*.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 388–397.

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

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

George H. John and Pat Langley. 1995. Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345.

S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. 2001. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation*, 13(3):637–649.

Fabian Pedregos, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Appendix A: Model Parameters

Three-input model parameters: 20k most frequent tokens per branch, maximum text length of 25, 1024 LSTM units per layer, 300-dimensional dense layers, batch size of 128, 15 training epochs, and the Adam weight optimization.

Two-output model parameters: 20k most frequent tokens, maximum text length of 100, 300 LSTM units, batch size of 128, dense layer sizes of 300 and 150 (respectively), 10 training epochs, and the Adam weight optimization.