

# ADAPTIVE VOCABULARY CONSTRUCTION FOR FRUSTRATION INTENSITY MODELLING IN CUSTOMER SUPPORT DIALOG TEXTS

Janis Zutens and Viktorija Leonova

Department of Computer Science, University of Latvia, Riga, Latvia

## ABSTRACT

*This paper examines emotion intensity prediction in dialogs between clients and customer support representatives occurring on Twitter. We focus on a single emotion type, namely, frustration, modelling the user's level of frustration while attempting to predict frustration intensity on the current and next turn, based on the text of turns coming from both dialog participants. A subset of the Kaggle Customer Support on Twitter dataset was used as the modelling data, annotated with per-turn frustration intensity ratings. We propose to represent dialog turns by binary encoded bags of automatically selected keywords to be subsequently used in a machine learning classifier. To assess the classification quality, we examined two different levels of accuracy imprecision tolerance. Our model achieved a level of accuracy significantly higher than a statistical baseline for prediction of frustration intensity for a current turn. However, we did not find the additional information from customer support turns to help predict frustration intensity of the next turn, and the reason for that is possibly the stability of user's frustration level over the course of the conversation, in other words, the inability of support's response to exert much influence to user's initial frustration level.*

## KEYWORDS

*Neural Networks, Emotion Annotation, Emotion Recognition, Emotion Intensity, Frustration*

## 1. INTRODUCTION

With the growing popularity of social networks and the exponential increase of user-generated content volume, automated language understanding is becoming ever more relevant. And emotion recognition plays no small part in this understanding. By their nature, humans are emotional beings, and emotions are very important for interpersonal communication. For this reason, many researchers have studied automatic emotion annotation, probably for as long as the machine learning field has existed. Most of these researchers have focused on variants of Ekman's emotion classification schema [1], annotating texts with several basic emotions. However, being interested in a specific task — namely, conversations between customers and customer support representatives — we concentrate on one specific emotion, frustration, and how it changes over the course of a dialog. The reason for this is that the main indicator of success for customer support is customer satisfaction or dissatisfaction, where dissatisfaction is captured by the emotion that we label as frustration.

In this work, we examine two hypotheses:

1. In customer support dialogs, the user's turn-by-turn frustration intensity can be predicted from the text of the user's message, and, in particular, from the presence of keywords – a

set of words (including also emojis and other non-lexical textual tokens) that correlate with specific frustration intensity levels.

2. In customer support dialogs, the frustration intensity of the user's current turn can be predicted from keywords in the user's previous turn together with keywords (from a different set) in the intervening turn from customer support. This targets the intuition that the manner in which the customer support representative responds to the user's utterances should have some effect on the user's emotional state going forward.

To test these hypotheses, we built a machine learning model and trained it on a dataset annotated specifically for this purpose, running a series of experiments as described in Section 5, Experiments and Results.

This paper is structured as follows: in Section 2 “Background and Related Work” we examine the previous works in the field of the emotion recognition and emotion intensity annotation, including the evolution of emotion in dialogs and available datasets. In Section 3 “Data Selection and Annotation” we explain how the dataset for training the model was selected and annotated. Section 4 “Frustration Intensity Prediction” explains the concept of frustration used in our research, the definition of frustration intensity and its evolution is given, and the main terms are introduced. Section 5 “Experiments and Results” provides the detailed description of conducted experiments, models constructed, and results achieved. In Section 6 “Discussion” we discuss the results provided in Section 5 and their interpretation. Finally, Section 7 “Conclusions and Future Work” gives a short summary of this work, results achieved and their possible development.

## **2. BACKGROUND AND RELATED WORK**

Virtually since the beginning of Machine Learning (ML) research, there have been attempts to apply ML to emotion annotation, first of speech (as the easier task, since speech signals carry additional, paraverbal information about the speaker's emotional state) and then also of text, as early as in 2005 by Alm et al. [2]. Most such researches used one or another version of Ekman's six emotion model [1]. Examples include Balahur et al., 2013 [3], Kao et al., 2009 [4] and others. With the development of social networks, the focus of work in emotion annotation has shifted toward emotion annotation in messages posted by users in social networks, such as Facebook, e.g. Al-Mahdawi and Teahan, 2019 [5], Weibo, e.g. Lee and Wang, 2015 [6] or Twitter, like Duppada and Hiray, 2017 [7], with Twitter being one of the most fruitful sources due to the open and concise nature of the posts it supports: short texts, sometimes accompanied by a picture or self-annotated with hashtags. Such self-annotations can even be used as the foundation for gold standard corpus labelling, as done by Gonzalez-Ibanez et al. in 2011 [8]. Several emotions have found their way into automated annotation, especially the basic emotions as identified by Ekman (fear, anger, joy, disgust, surprise and sadness), as for example Badaro et al., 2019 [9]. And even such elusive notions as sarcasm and irony have been researched, for example by Reyes et al., 2013 [10]. Frustration, however, has not been widely researched. There have been a few papers focusing on frustration, such as Klein et al., 2006 [11], or Hone, 2002 [12], but not many. Hu et al., 2018 [13] discuss the correlation between the emotional tone of customer support messages and user messages, and the tones they study include frustration among others. We believe that, especially in the field of business communication, automatic frustration recognition targets a relatively unaddressed need.

Whereas much earlier work sought primarily to output binary, categorical labels (predicting the presence or absence of specific emotions), labelling and predicting gradations of emotion

intensity is only recently becoming more widespread. Examples include Goel et al., 2017 [14], Bravo-Marquez et al., 2019 [15], and Badaro et al., 2019, analysing emotion intensity in tweets and providing a Weka package for automatically annotating tweets with intensities ratings for anger, fear, joy, and sadness. However, as there has been little work on frustration recognition in general, automatic recognition of frustration intensity mostly remains unaddressed — one exception being the aforementioned Hu et al., 2018, who annotated and modelled intensities for 8 differing emotional “tones” (or language production styles): anxious, frustrated, impolite, passionate, polite, sad, satisfied, and empathetic; our work differs from theirs in that we focus exclusively on frustration, while they explore correlations between the user’s vs. the support agent’s tone for all pairwise combinations of these 8 tones. Their work and ours also differ in the methods used for selecting keywords associated with a given tone or emotion, and in the architecture and goal of the machine learning models developed. Whereas they train a seq2seq model (sequence-to-sequence, using a recurrent neural network) for generating dialog responses with specified tones, we develop relatively simpler neural models for predicting user frustration gradations given previous user + support agent turns (their analysis of correlations between user vs support agent tones is carried out via linear regression.)

While there are several publicly available dialog datasets, for example Taskmaster-1 [16] or DailyDialog [17], none have directly addressed the modelling of participants’ turn-to-turn emotional state dynamics in a goal-oriented context, to the best of our knowledge. With respect to dialog datasets and research on automated dialog agents (or “chatbots”) an important distinction is often drawn between goal-oriented dialog agents (where the user is seeking to accomplish some task with assistance from the automated agent) vs. free-chat agents (which attempt to simulate human-style conversations with users, “chatting” with no specific goal other than entertainment, or, possibly, some kind of therapeutic objective). The labelling and structure of the datasets associated with each of these chatbot types are, in general, very different. (Taskmaster and DailyDialog are prototypical examples of datasets for goal-oriented vs free-chat agents, respectively). In one case, the primary focus is on identifying the user’s ‘intent’ (what she is trying to achieve) and shaping further interactions to elicit whatever additional information might be required to complete it. Free-chat agents, on the other hand, are mostly concerned with generating responses that simulate what a human conversational partner might say in the same situation. The free-chat setting is where most previous research on identifying emotions and generating responses with emotionally appropriate language has been done.

Customer support agents can be viewed as a hybrid of goal-oriented and free-chat agents, in that the client usually does have a specific objective (resolving or at least reporting a specific problem), but emotional dynamics are also very important: in the final analysis, the primary objective of the dialog agent can be formulated as an emotional state (“client satisfaction”). Automated goal-oriented dialog agents have been studied in quite a few works, for example Ham et al., 2020 [18], as have affect-driven free-chat dialog agents e.g. Colombo et al., 2019 [19], and Lubis et al., 2018 [20], focusing on providing affect-sensitive responses, but very few works have investigated dialog agents that attempt to address both concerns simultaneously [21], [13].

### **3. DATA SELECTION AND ANNOTATION**

We have examined several publicly available potential sources of data for our research, but unfortunately, none were fully satisfying the requirements of our experiment: either they were not goal-oriented or provided no information about the underlying emotional state. The example of not goal-oriented dialogs is provided in Figure 1, where the conversation is sourced from DailyDialog.

A: I'm **worried** about something.

B: What's that?

A: Well, I have to drive to school for a meeting this morning, and I'm going to end up getting stuck in rush-hour traffic.

B: That's **annoying**, but nothing to worry about. Just breathe deeply when you feel yourself getting upset.

A: Ok, I'll try that

Figure 1. Example of a not goal-oriented dialog

Another type of available corpora is goal-oriented conversations, that lack emotions. For example, one such dataset is Taskmaster-1, and Figure 2 provides a characteristic dialog from the dataset.

"text": "Hi, I'm looking to book a table for Korean food."

"text": "Ok, what area are you thinking about?"

"text": "Somewhere in Southern NYC, maybe the East Village?",

"text": "Ok, great. There's Thursday Kitchen, it has great reviews."

Figure 2. Example of a dialog with no information about user's emotional state

After having studied the available conversation datasets, we have deemed those unfitting for our purposes and have decided to use a subset of the Kaggle Customer Support on Twitter dataset<sup>1</sup>, additionally annotated for frustration intensity, as a basis for our studies. Nowadays customer support using such channels as social networks like Twitter and Facebook gains increasing popularity, and the measure of effectiveness of customer support is essentially a satisfaction, or, more precisely, however, probably, less customary, a reduction of customer's frustration, and providing the system with a possibility to automatically evaluate a customer's frustration level, and in the ideal scenario, providing the automated agent with a way to increase customer satisfaction, in other words, to reduce customer's frustration by tailoring dialog responses appropriately, would definitely bring a business value to companies.

As the Kaggle Customer Support dataset is not organized in any way, and represent the collection of tweets originating from, replying to or quoting customer support accounts, the first step we took was organizing this collection into threads, by linking reply ids to tweet ids, thus forming consecutive dialogs. The resulting set of conversations was then filtered so that the final version would only contain the dialogs that fulfil the criteria. First, we have excluded the conversation with more than one participating user, as we were about to study the effect of support's turn on the frustration intensity of the user, and it is impossible to do so when there is another intervening party. For the same reason, we were only leaving in the conversations that had at least two user's turns with at least one support's turn in between. This way, we were able to determine, how user's emotional state has changed after support's turn, and only that, to the information available to us.

After the conversations fulfilling the requirements were assembled from the dataset in the described way, a relatively small subset of dialogs was selected from those for annotation. As the literature study suggests [22], experiments involving automated emotion annotation with the help of neural network-based models typically use a number of tweets that varied around several hundred entries. Thus, we have decided to follow this example and annotate four hundred dialogs, of no less than eight hundred customer turns. As to the selection of dialogs, due to the

fact that original dataset was not organized in any specific way, we concluded that choosing first four hundred would provide us with effectively random sample, while at the same time ensuring the possibility to extend the dataset when required by annotating and appending subsequent dialogs.

After the conversations were selected for final dataset, they were anonymized and unified in the following way: first, the user Twitter ids was replaced with “USER” label, while support Twitter id was replaced with “SUPP” label. The tweet ids and reply ids were removed altogether, and if a dialog contained several user or support messages in uninterrupted sequence, they were joined together. This way, the user’s and support’s turns always followed one another in the following manner: USER -> SUPP -> USER -> SUPP ->.... The private and sensitive information, such as email addresses, had been already taken care of and was replaced in the Kaggle Customer Support dataset by generic placeholders like “\_\_email\_\_”.

The selected dialogs were assigned unique ids and together with the instruction on their annotation were sent to three independent annotators. The instruction asked them to place a single symbol next to each of the customer turns. This symbol could take six values. First five were integers from 0 to 4, denoting a customer’s frustration level as perceived by the annotator, where zero was to denote that the customer is satisfied or is in a neutral emotional state, while four indicated ultimate frustration. Besides numerical values, the instruction has asked the annotators to mark the turn with “n”, if they could not make a conclusion about the customer’s emotional state based on the message, e.g. in the case of giving single-word answers or providing purely technical information in response to a question, such as stating the customer number or address. The value could also have been left empty, which meant that the annotator could not interpret the message, for example, in case if the language was unknown to him or the text was not comprehensible in some other way.

After the annotated files were received back from the individual annotators, they were combined into a single master file, in which every user turn in every conversation was associated with three assigned values, one from each annotator (note that some of these values could be ‘n’ or blank, as previously described). Then the file was further filtered to exclude the dialogs that did not have all three numeric values for each of the customer’s turns, that is the dialogs that did not contain any “n” or missing values — yielding a total of 376 dialogs, with an average dialog length of 5.2 turns. For our experiments we have calculated the median values for each dialog turn, which allowed us to have a value representing the annotators consent while keeping the values discrete, thus providing for the application of classification. We have also calculated the “deltas” — difference between the consecutive turns’ frustration intensity values and have noticed some consistent patterns.

First, major transitions for the worse (+2 delta in frustration rating) seem often to be situations where the customer support representative tells the user to do something that he or she has already tried. This, presumably, is something that the customer support representative had no way of knowing and is certainly not part of the information available in the input data for our model (which sees only the text of the preceding turns of the dialogue). Another pattern we noted is when the user was probably in fact more frustrated than his first question or statement suggests, but expresses his full frustration only in a subsequent turn. Once again, this is not something that the customer support agent (or a machine learning model) is likely to be able to anticipate.

Outlier transitions for the better (-3 or -4 delta in frustration rating), on the other hand, in general seem to be due to something that has happened in the real world (as opposed to in the dialog) to resolve the user’s complaint (e.g. the user found a way to resolve it themselves, or the problem got otherwise resolved in the meantime).

**TWCS-T1894 (delta: +2)**

USER:	call centre has the worst customer service ever. keep you on hold for endless hours without an apology. Mum is not happy!	4
SUPP:	We're disappointed to hear this Stephen. Is there something you would like our assistance with? ^Kit	
USER:	Can you please contact __email__ and can give you further details please	2
SUPP:	Hi Stephen. For security please remove the email address from the public feed. We're unable to contact passengers by email. 1/2 Is your mother able to contact us directly? ^Davina 2/2	
USER:	She's been trying to do that, that's the point! You should be contacting her as she can't get through to you via phone or email	4
SUPP:	Hi Stephen, follow and DM us more info along with your mum's contact details. ^Helen	

**TWCS-T3076 (delta: +2)**

USER:	why don't your marketing emails have an unsubscribe link?	2
SUPP:	Hello, there should be a link at the bottom of the emails. Please DM and we can discuss further. Thank you. ~RD	
USER:	Hidden fees, false marketing, spam, crappy customer service, increasing costs to reward loyalty. Done with your junk.	4
SUPP:	I will be happy to assist, please DM me for assistance. -GR	

**Turns for the better: frustration rating decreases (-3 or -4 points)**

**TWCS-T1691 (DELTA: -4)**

USER:	I legitimately spent an hour trying to deal with USPS cause I had 1 question and they just hung up on me or wasn't any help, I could have saved my fucking time by just checking my mailbox because sure enough I got the UPS letter saying my package was in oh my gOD	4
SUPP:	Is there something that we can assist you with? DM our team ^WS <a href="https://t.co/wKJHDXWGRQ">https://t.co/wKJHDXWGRQ</a>	
USER:	Nope, I've got my package thanks	0

**TWCS-T156 (delta: -3)**

USER:	So frustrated with @ChipotleTweets ☹️ Ordered dinner on Saturday using their app. Order was wrong AND they charged my credit card twice	3
SUPP:	That's concerning. Please share this with <a href="https://t.co/nrhUDiEk7G">https://t.co/nrhUDiEk7G</a> . -Shawn	
USER:	Thank you @ChipotleTweets for resolving my issue so quickly!! Y'all are the best ☺️ #fanforlife	0
USER:	So frustrated with @ChipotleTweets ☹️ Ordered dinner on Saturday using their app. Order was wrong AND they charged my credit card twice	3

**4. FRUSTRATION INTENSITY PREDICTION**

This section describes the proposed approach of using automatically selected keywords in frustration intensity prediction for dialogs.

#### 4.1. Review and Data Preparation

This research focuses on frustration intensity prediction for user-side turns in Twitter-originated customer support dialogs, where the turns are represented by binary encoded bags of automatically selected keywords, and, correspondingly, our two research tasks are:

1. Predicting frustration intensity for the actual user’s turn,
2. Predicting frustration intensity for the next user’s turn.

The goal of the experimentation is to show that the user’s frustration intensity can be efficiently predicted from the automatically selected keywords found in the text.

The outline of the proposed approach (see also Fig. 3):

1. Extract and rank keywords from the annotated dataset,
2. Choose a certain number of the best keywords and encode dialog texts using them,
3. Construct the prediction model.

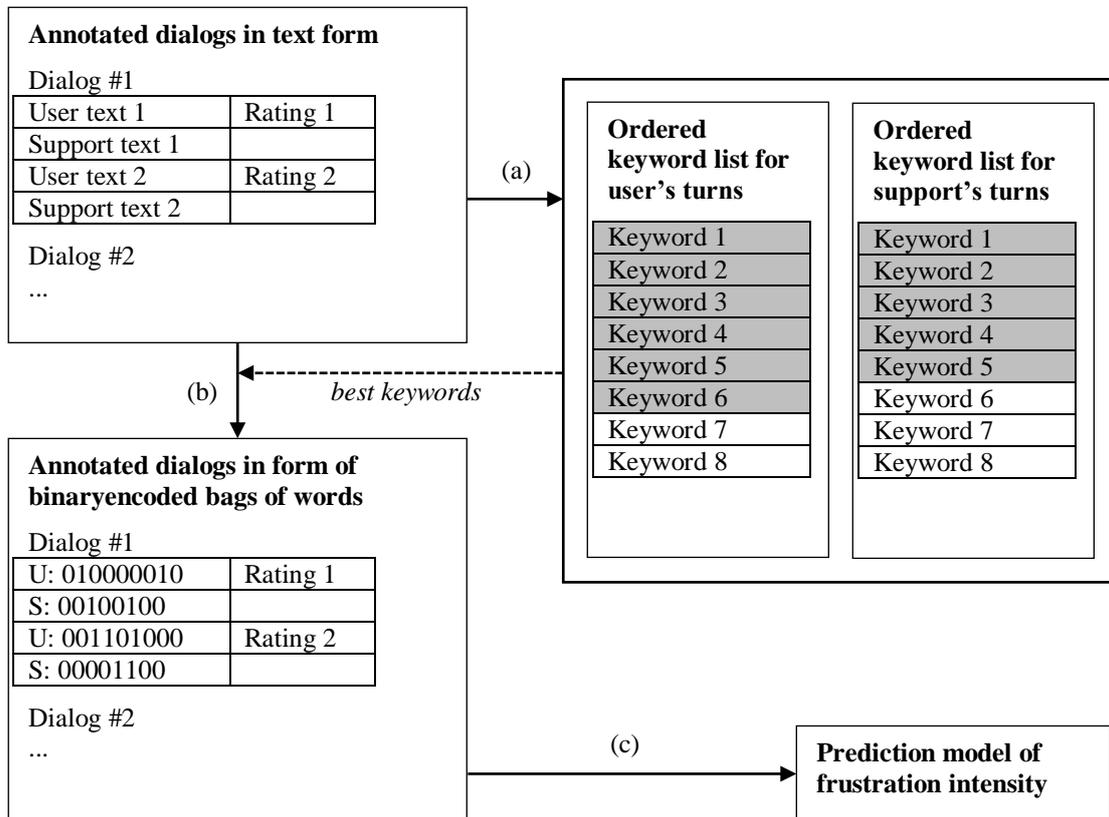


Figure 3. Method overview for the frustration intensity prediction: a) keyword extraction and ranking, b) dialog encoding, c) model construction.

For our experiments, we used a corpus of 376 dialogs having an average dialog length of 5.2 turns (counting both user and support turns, i.e. three user turns and two intervening support turns). Each user’s turn was rated by three annotators, and for the experimentation purpose we utilized only the “valid” turns — the ones that received a numeric value by all three experts. Some turns were difficult to evaluate (see Section 3), so the final version of the data consisted of

843 annotated valid user's turns (see Table 1), as well as 470 valid support' turns. By a valid support's turn we here understand the one between two valid user's turns. As for the frustration dynamics in the dialogs, the average frustration intensity change from one user's turn to the next one was -0.35, which means that, in general, frustration intensity was observed to decrease from turn to turn, but only slightly. Over the course of a short dialog, the user's frustration intensity rating, on average, was expected to remain essentially unchanged.

Table 1. Overview of user's turns used in experiment data.

<b>Frustration intensity</b>	<b>Amount of valid user's turns</b>
0	155
1	125
2	234
3	239
4	90
TOTAL	843

For the turn's text encoding, we have constructed two keyword vocabularies as ordered lists of lower-cased tokens that occurred in the valid turns for at least three times, separately for the user's turns and support' turns:

- for the user's turns, ranked by standard deviation of ratings of the turns a particular token occurred in,
- for support's turns, ranked by standard deviation of the difference of ratings of the two neighbouring user's turns with respect to the turns a particular token occurred in.

Thus, we have obtained two vocabularies:

- user's keyword vocabulary  $V_{\text{user}}$  of size 941,
- support's keyword vocabulary  $V_{\text{supp}}$  of size 450.

Individual dialog turns for the experiments were represented in bag-of-words encoding using the best keywords from the vocabulary:

- encoding was done separately for the user's turns and support's turns,
- we used different number of 'best keywords' in different experiments: the number of keywords was defined by the actual experiment configuration.

## 4.2. Quality Measures and Experiment Tasks

For evaluation of the prediction model, we used accuracy metrics. As the prediction classes (0..4) are ordered, we also introduced a weaker evaluation metrics — accuracy with tolerance +/- 1 (so that an "off-by-one" prediction is also considered correct). Accuracy with tolerance seems more adequate for these tasks (e.g. predicting 2 when the "correct" rating is annotated as 3, is not equally wrong to predicting 0 in the same situation).

To demonstrate the proposed approach, we carried out experiments on two different research tasks:

1. Predicting frustration intensity for the actual user's turn,

2. Predicting frustration intensity for the next user’s turn.

For both experiment tasks, we also describe the baselines to evaluate the obtained models.

Table 2. Baselines for the experiment task #1 (constant rating values from the rows marked in grey).

Rating, r	Amount of valid user’s turns, exact accuracy		Amount of valid user’s turns, accuracy with tolerance 1	
	Count, c <sub>r</sub>	Percentage, p <sub>r</sub>	Count, c <sub>r-1</sub> + c <sub>r</sub> + c <sub>r+1</sub>	Percentage, p <sub>r-1</sub> + p <sub>r</sub> + p <sub>r+1</sub>
0	155	18.4%	280	33.2%
1	125	14.8%	514	61.0%
2	234	27.8%	598	70.9%
3	239	28.4%	563	66.8%
4	90	10.7%	329	39.0%
TOTAL	843	100%	843	843

**Experiment Task #1: Frustration intensity prediction for the actual user’s turn (see Fig. 4).**

Task description (see Fig. 4):

- Predict the frustration intensity of a user’s turn, given the turn represented by binary encoded bags of automatically selected keywords.

The baseline (see Table 2) — can we predict better than simply choosing the most frequent rating for all input data:

- Constant value 3 if the exact accuracy is used as metrics,
- Constant value 2 if the accuracy with tolerance 1 is used as as metrics;

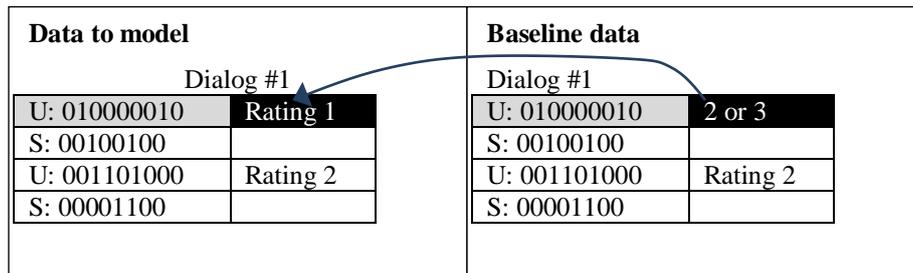


Figure 4. Modelling actual frustration intensity prediction (input marked with grey, output marked with black).

**Experiment Task #2: Frustration intensity prediction for the next user’s turn (see Fig. 5).**

Task description (see Fig. 5):

- Predict the frustration intensity of the next user’s turn, given current user’s turn and the following support’s turn, both represented by binary encoded bags of automatically selected keywords.

Baseline (see Fig. 5) – can we predict the next rating better than exactly taking the predicted rating of the current turn also for the next turn:

- Predicted frustration of the current user’s turn using our model for task #1 solving.

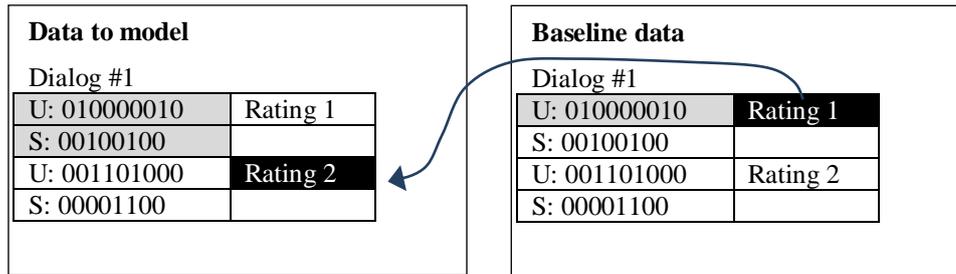


Figure 5. Modelling next frustration intensity prediction (input marked with grey, output marked with black).

## 5. EXPERIMENTS AND RESULTS

### 5.1. Experiment Configuration and Flow<sup>2</sup>

The experimentation was carried out in two phases:

1. Preparation:
  - a. hyperparameter search for neural network models,
  - b. selection of the set input configurations to represent dialog data (as shown in column #1 of Tables 1 and 2);
2. Main run – run experiments with selected hyperparameters and with every selected input configuration.

#### 5.1.1. Experiment Preparation

For each of the two experiment tasks, a hyperparameter search covering several thousand experimental runs was conducted, to select a final model consisting of a multi-layered perceptron with one hidden layer with the following final configuration:

1. Input: binary input of several hundreds of values (as per Section “Method Overview and Data Preparation”) representing the text of one or two dialog turns — as amounts of keywords per turn:
  - a. for experiment task #1 we have selected the following input configurations: 50, 100, 300, 500 (see Table 1) for the number of keywords to represent a user’s turn,
  - b. for experiment task #2 we have selected the following input configurations: 50/50, 200/100, 500/200, 700/350, meaning that to train the baseline model input configurations 50, 200, 500, 700 were used respectively (see Table 2) —the number of keywords to represent the current user’s turn/the number of keywords to represent the following support’s turn;
2. Hidden layer: 64 neurons;
3. Output: categorical of 5 possible values representing frustration intensity (0..4);

The number of epochs for each experiment: 50.

### 5.1.2. Experiment Main Run

With the obtained model architecture, we have conducted a further series of experiments using a different input — encoding configurations as selected in the preparation phase.

For each input configuration of each of the two experiment tasks, we used Leave-one-out cross-validation to evaluate the model for the average accuracy (see Section “Quality Measures and Experiment Tasks”). Experimentation for a fixed input configuration consisted of the following steps:

- For all annotated  $n$  data points in the dialog dataset relevant to the experiment (as for Section “Quality Measures and Experiment Tasks”):
  - Prepare the data for the proposed (target) model:
    - the current data point is reserved for testing:
      - for task #1 – a data point is one user’s turn in a dialog to predict the current intensity (as in Fig. 4),
      - for task #2 – a data point is the current user’s turn, as well as the following support’s turn to predict the next intensity (as in Fig. 2);
    - the rest of  $n-1$  data points go for training;
  - Prepare the data for the baseline:
    - for task #1, a fixed baseline value is used – the most common label in the dataset (Baseline columns in Table 3),
    - for task #2, separate data for the baseline model are prepared (current user’s turn only);
  - Train the models for 50 epochs:
    - for task #1, only the target model is trained (as the baseline is fixed),
    - for task #2, the baseline model is also trained;
  - Collect the experiment results:
    - For task #1 – apply the model to the test data and collect accuracy measurements (Result columns in Table 3),
    - For task #2 – apply both models to the test data and collect accuracy measurements:
      - target model accuracy (Result columns in Table 2),
      - baseline accuracy (Baseline columns in Table 2).

Evaluate the input configuration: the result is the average accuracy of the  $n$  models of the input configuration (as obtained using Leave-one-out cross-validation).

## 5.2. Experiment Results

When running our series of experiments, we found that the results for repeated runs using a given configuration generally varied only within a range of one percent, so here we report all results rounded to whole numbers (see Tables 3 and 4).

**Experiment Task #1: Frustration Intensity Prediction.**

Table 3. Results for frustration intensity prediction.

Input configuration: best user keyword count	Accuracy, %		Accuracy with tolerance 1, %	
	Result	Baseline	Result	Baseline
50	37	28	74	71
100	41		78	
300	41		80	
500	41		80	

Experimental results show that:

- Frustration intensity can be effectively predicted from the presence of selected keywords;
- 100 keywords can be sufficient for predicting the frustration with the ‘exact’ accuracy (with no tolerance);
- Using more keywords gives better results for accuracy with tolerance.

**Experiment Task #2: Frustration Intensity Dynamics Prediction.**

Table 4. Results for frustration intensity dynamics prediction.

Input configuration: user keyword count / support keyword count	Accuracy, %		Accuracy with tolerance 1, %	
	Result	Baseline	Result	Baseline
50/50	34	28	58	67
200/100	34	30	62	70
500/200	34	33	68	70
700/350	30	31	65	69

Experimental results show that:

- Frustration intensity for the next turn can be to some extent predicted from presence of selected keywords in the user’s actual turn (baseline model);
- Using additional keywords from the customer support turn doesn’t improve the predictions.

**6. DISCUSSION**

In this work, we have constructed a neural network-based model for predicting user frustration intensity from the text of a user tweet addressed to a customer support. This model takes an encoded representation of the user message as an input and gives an output in the form of an integer rating of frustration intensity on a 5 point scale (0 to 4), achieving a precision of 41%, 14% higher than a baseline which simply assigns the most frequent label to all instances. In addition to exact precision, we also calculate precision with tolerance (allowing a difference of 1 between the actual and predicted rating). Using this “+/-1 accuracy” metric, our model achieves 80%, 9% higher than the baseline (71% using this metric). This allows us to say that to a certain degree frustration intensity can be predicted from the text of a user's message precisely, and in 80% of cases it can be predicted approximately.

In addition, we have constructed another neural network-based model that predicts the user's emotional state dynamics from the contents of the support agent's reply to a preceding message from the user. From encoded representations of the user's message and the support agent's message, it attempts to predict the frustration rating that annotators assigned to the next (user's) turn. As the baseline model we have used the prediction of the frustration intensity for the initial user message, under the assumption that the user's frustration remains unchanged. The achieved precision was 34%, a very slight (1%) improvement over the baseline. Also, for this scenario, allowing +/- 1 tolerance in the predicted frustration intensity doesn't improve over the baseline (just using the prediction for the initial message is better), thus implying that knowing the contents of the support agents message provides no additional useful information toward predicting changes in the user's state of frustration (and which, in general, does not significantly change from one turn to the next).

We have already noted the overall tendency for the user's level of frustration tends to remain mostly unchanged from turn to turn. We hypothesize that this might be at least partially explained by the fact that customer support representatives are already formulating their replies with the goal of trying to reduce, or at least to not increase, the customer's frustration or level of dissatisfaction with their company's products or services (they are, in fact, often trained and explicitly motivated to do so).

Manually examining our data in more detail, we find only 7 examples of dialogues where the user's level of frustration has been labelled as changing for the worse by more than 1 point from one turn to the next (in all such examples the increase is +2 points; there are no examples of a jump of +3 or +4 points). A change in rating for the better is relatively more common: there are 44 examples of turn-to-turn transitions with a -2 delta (where the user's level of frustration has decreased by two points), 13 with -3, and one with -4 (which would mean that the user started out maximally frustrated/dissatisfied but transitioned to being completely satisfied within a single dialog turn). Some examples of such exceptional dialogs are given in Section 3.

But such outlier transitions are the exception rather than the rule — the overall finding in terms of turn-to-turn dynamics is well illustrated by the relatively strong performance of our baseline model, which simply assumes that the user's frustration level will remain unchanged from the previous turn.

## **7. CONCLUSIONS AND FUTURE WORK**

In this paper, we have presented a new dataset — a subset of the Kaggle Twitter Customer Support dialogs consisting of close to 400 dialogs and comprising almost 900 individual customer tweets, annotated for frustration intensity on the scale of 0 to 4. We have selected the most popular grade as a baseline and demonstrated that frustration intensity can be predicted based on the contents of an individual tweet with an accuracy significantly higher than the baseline (41% compared to 27%). This result was achieved by constructing a neural network and training a simple classification model. We also examined the effect of customer support turns on the emotional state of the user and found that, typically, the user's emotional state mostly remains unchanged, with a small decrease of 0.34 points on average from one turn to the next. Currently, in contrast to our generally positive finding for predicting turn-by-turn frustration ratings from text-based features, we conclude that, given the challenges in precise calibration of the user's frustration level — due at least partially to the subjective and fleeting nature of the emotion itself and the difficulty of estimating it by a third party purely from the text of a conversation, trying to model this dynamic as a function of the emotional valence of the support agent's messages doesn't yield any strong results (at least not using classification models like the neural models we tried).

In the future, we are looking towards possibly adapting and applying this methodology to dialogs in Latvian, Latvian being a low-resource language where practically no work on automatic emotion annotation with machine learning methods has been undertaken, and analysing the effect of another language on the accuracy of automatic annotation of frustration level, and on the feasibility of predicting the dynamics of the user's emotional state.

## ACKNOWLEDGEMENTS

The research has been supported by the European Regional Development Fund within the joint project of SIA TILDE and University of Latvia "Multilingual Artificial Intelligence Based Human Computer Interaction" No.1.1.1.1/18/A/148.

## REFERENCES

- [1] P. Ekman, (1992) "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3-4, pp.169–200.
- [2] C. O. Alm, D. Roth, and R. Sproat, "Emotions from text," *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT 05, 2005*, pp.579–586.
- [3] A. Balahur, J. M. Hermida, A. Montoyo, and R. Muñoz, (2013) "Detecting implicit expressions of affect in text using EmotiNet and its extensions," *Data & Knowledge Engineering*, Vol. 88, pp.113–125.
- [4] E. C.-C. Kao, C.-C. Liu, T.-H. Yang, C.-T. Hsieh, and V.-W. Soo, (2009) "Towards Text-based Emotion Detection A Survey and Possible Improvements," *2009 International Conference on Information Management and Engineering*, pp.70-74.
- [5] A. Al-Mahdawi and W.J. Teahan (2019) "Automatic emotion recognition in English and Arabic text" (Doctoral dissertation, Bangor University).
- [6] S. Lee and Z. Wang, (2015) "Emotion in Code-switching Texts: Corpus Construction and Analysis," *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pp.91-99.
- [7] V. Duppada and S. Hiray, (2017) "Seernet at EmoInt-2017: Tweet Emotion Intensity Estimator," *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp.205-211.
- [8] R. González-Ibáñez, S. Muresan, and N. Wacholder, (2011). Identifying sarcasm in Twitter: a closer look. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.581-586.
- [9] G. Badaro, H. Jundi, H. Hajj, and W. El-Hajj, (2018) "EmoWordNet: Automatic Expansion of Emotion Lexicon Using English WordNet," *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp.86-93.
- [10] A. Reyes, P. Rosso, and T. Veale, (2012) "A multidimensional approach for detecting irony in Twitter," *Language Resources and Evaluation*, Vol. 47, No. 1, pp. 239–268.
- [11] J. Klein, Y. Moon and R.W. Picard, R.W., (2002). "This computer responds to user frustration: Theory, design, and results." *Interacting with computers*, Vol. 14, No. 2, pp.119-140.

- [12] K. Hone, (2006). "Empathic agents to reduce user frustration: The effects of varying agent characteristics." *Interacting with computers*, Vol. 18, No. 2, pp.227-245.
- [13] T. Hu, A. Xu, Z. Liu, Q. You, Y. Guo, V. Sinha, J. Luo, and R. Akkiraju, (2018) "Touch Your Heart," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI 18*, pp.1-12.
- [14] P. Goel, D. Kulshreshtha, P. Jain and K.K. Shukla, (2017) Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets. *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 58-65.
- [15] F. Bravo-Marquez, E. Frank, B. Pfahringer, and S. M. Mohammad, (2019). Affective tweets: A weka package for analyzing affect in tweets. *Journal of Machine Learning Research*, Vol. 20, No. 92, pp.1–6.
- [16] B. Byrne, K. Krishnamoorthi, C. Sankar, A. Neelakantan, B. Goodrich, D. Duckworth, S. Yavuz, A. Dubey, K.-Y. Kim, and A. Cedilnik, (2019) "Taskmaster-1: Toward a Realistic and Diverse Dialog Dataset," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- [17] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, (2017) *Dailydialog: A manually labelled multi-turn dialogue dataset*. arXiv preprint arXiv:1710.03957.
- [18] D. Ham, J. G. Lee, Y. Jang, and K. E. Kim, (2020) End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp.583–592.
- [19] P. Colombo, W. Witon, A. Modi, J. Kennedy, and M. Kapadia, (2019) "Affect-Driven Dialog Generation," *Proceedings of the 2019 Conference of the North*, pp. 3734–3743
- [20] N. Lubis, S. Sakti, K. Yoshino, and S. Nakamura, (2018). Eliciting positive emotion through affect-sensitive dialogue response generation: A neural network approach. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp.5293–5300.
- [21] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, (2017) "A New Chatbot for Customer Service on Social Media," *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 3506–3510.
- [22] Leonova V., (2020) "Review of Non-English Corpora Annotated for Emotion Classification in Text." *International Baltic Conference on Databases and Information Systems*. pp. 96-108.

## AUTHORS

**Jānis Zuters** received the PhD degree in Computer Science in 2007. His research interests include machine learning, neural networks, and natural language processing. Since 1999, he has been with the University of Latvia, Faculty of Computing (since 2017 as a Professor).



**Viktorija Leonova** is a PhD student in the University of Latvia, Computer Science Department. Acquired M.Sc. in Computer Science in Open University of Cyprus in 2015.

