Learning From Personal Longitudinal Dialog Data

Charles Welch, Verónica Pérez-Rosas, Jonathan K. Kummerfeld, and Rada Mihalcea
University of Michigan

***Abstract*—We explore the use of longitudinal dialog data for two dialog prediction tasks: next message prediction and response time prediction. We show that a neural model using personal data that leverages a combination of message content, style matching, time features, and speaker attributes leads to the best results for both tasks, with error rate reductions of up to 15% compared to a classifier that relies exclusively on message content and to a classifier that does not use personal data.**

Most dialog research provides an overall view of speakers' language and interaction behaviors based on data from recorded spoken conversations, movie scripts, social network messaging, forums, instant messaging, and audio subtitles.[1–5] These corpora contain a diverse set of speakers. Thus, the developed models are not tailored to individual speakers, who might have preferences and behaviors different than the consensus trends.

In this article, we address discourse analysis in personal dialog data. In particular, we seek to explore what can be learned from personal messaging history by analyzing language usage and communication patterns. We conduct our analysis over a large set of conversations obtained from the instant messaging history of several individuals. The conversation set contains 1.3 million messages from a five-year time span. We label speaker social relations using seven categories—gender, school, work, relationship status, family, age, and cultural background. We then use psycholinguistic-inspired analysis to analyze language usage within groups in these categories. We use the insights from these analysis to derive features that represent the message content, messaging frequency, and messaging timing. We also derive several features to capture interaction behaviors, including word usage and language matching across conversational groups. We use these features in combination with standard word embeddings to conduct two classification tasks: 1) predicting the next message in the conversation (based on the most common utterances); and 2) predicting the message response time. For both

Digital Object Identifier 10.1109/MIS.2019.2916965
Date of current version 19 September 2019.

1541-1672 © 2019 IEEE

tasks, models with our features and trained on personalized data perform best.

RELATED WORK

Speaker behavior in instant messaging services has been widely studied for tasks such as dialog act tagging and discourse analysis. Studies have attempted to classify messages into actions, such as “greet,” “accept,” or “reject” for online messaging, customer service interactions, and many other settings.[6,7] Other recent work has focused on the understanding of speakers, detecting the emotion they are expressing[8] and the relationship between speakers.[9] Holmer applied discourse structure analysis to identify and visualize message content and interaction structures.[10] Tuulos *et al.* inferred social structures in conversations, using heuristics based on participant’s references, message response time, and dialog sequences.[11] They represented the social structure using graph-based methods and explored features extracted from the graph to identify topics.

Many other dialog corpora exist. Recent work on building task-oriented and end-to-end dialog systems has used corpora from Twitter[12] and specific types of chatrooms, such as the Ubuntu chat corpus.[13] The construction of such datasets is motivated by the desire to have more useful dialog systems. Although much can be learned from these corpora, systems often also require commonsense reasoning to be effective.[14] The most relevant corpus for our work is the NUS SMS corpus, which contains publicly released text messages, however the authors could not collect messages received, restricting their analysis.[15]

DATASET

To enable our experiments, we invited individuals to contribute their personal messaging history for a study on personal longitudinal data. The study was approved by the Institutional Review Board (IRB) at the University of Michigan. To ensure data privacy, we recruited participants who could run our code on their own computers, keeping message content private and sharing only aggregate statistics with us. We recruited eight participants and provided them with detailed instructions on how to prepare the data and run the scripts.

We define the following conversation units: A *message* consists of all the text written by a participant in a conversation right before they press the send key. A *turn change* occurs when the author of the current message differs from the participant in the previous message. Note that a turn can be composed of multiple messages. We define a *conversation* as a sequence of turns between two individuals. *Message response time* is the amount of time that has passed between a message from a user and the previous turn change. On these platforms, conversations continue indefinitely, but shifts in response time can indicate when a synchronous exchange has ended.

Table 1. Distribution of messages and tokens (words, punctuation, emoticons) in conversations.

	Participant	Other	All
Total Messages	690,767	647,026	1,340,338
Average Unique Messages	63,039	62,907	123,568
Total Tokens	4,992,575	5,069,745	10,062,320
Average Unique Tokens	19,023	23,265	32,195
Average Tokens/Message	7.23	7.83	7.52

Unique averages are computed at the participant level.

All messages sent and received by participants via Google Hangouts, iMessage, and Facebook Messenger are considered, covering a range of short message service systems. The data spans a decade and contains about 1.3 million messages (there may be some overlap if participants spoke to each other though we cannot quantify it because we do not have access to the raw data), but we focus on a five year span containing the majority of messages: 2012 to 2017. We also exclude multiparty conversations and conversation partners with fewer than 100 messages. This leads to a final set of 508 interlocutor pairs and contains all the messages from conversations held between the participants and other individuals during 2012–2017. Table 1 shows corpus statistics. The data contain slightly more sent messages than received, but sent messages are slightly shorter.

Annotation of Social Interaction Categories

To enable our analysis, each participant manually labeled their conversation partners with

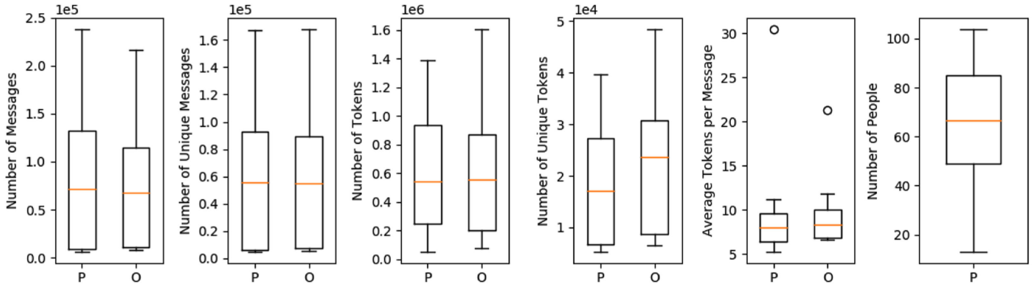

Figure 1. Distribution of number of messages and tokens between the (P) participants and their conversation partners (O) in our dataset.

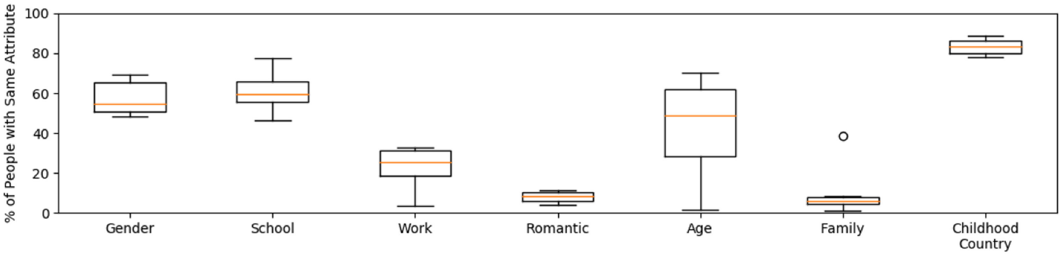

Figure 2. Shared attributes between participants and their conversation partners.

seven attributes that describe their social relationship. We chose attributes that they were likely to know about the people they converse with and may impact the way they write. The attributes are defined as follows:

Same Gender: The participant is the same gender as the other speaker.

School: The participant and the other speaker met while attending school.

Work: The participant and the other speaker know each other from work.

Romantic: The participant and the other speaker were in a romantic relationship at some point.

Family: The participant and the other speaker are related.

Relative Age: The participant is older, younger or the same age (± 1.5 years) as the speaker.

Childhood Country: The participant grew up in the same country as the other speaker.

These attributes and their values are used during the analysis and experiments presented throughout this article. We analyze aggregate statistics of our corpus including total messages and tokens exchanged, the distribution of attributes, and message production across time.

Message and Speaker Distributions

Figure 1 shows the distribution of messages and tokens across participants. The leftmost plot shows participants had from a few thousand to a few hundred thousand messages. Distributions are similar for participants and their partners across the number of messages, tokens, and unique messages. The distribution of unique tokens differs, providing some evidence for variation in writing, as each value for O is based on a set of individuals, while each value for P is based on one individual (the participant). The average number of tokens per message ranges from 5–12 with the exception of one outlier, whose messages were significantly longer.

We also examine the distribution of speaker attributes over conversation partners and across participants. Figure 2 shows this by representing the values "yes" (school, work, romantic, family) or "same" (gender, age, childhood country) for each attribute. For instance, the gender plot shows that the median proportion of conversation partners of the same gender as the participant is 54%. Note that while *age* takes three values the plot shows only *Relative Age* = same. The range is similar for older conversation partners but ranges from 11–37% for those who are younger.

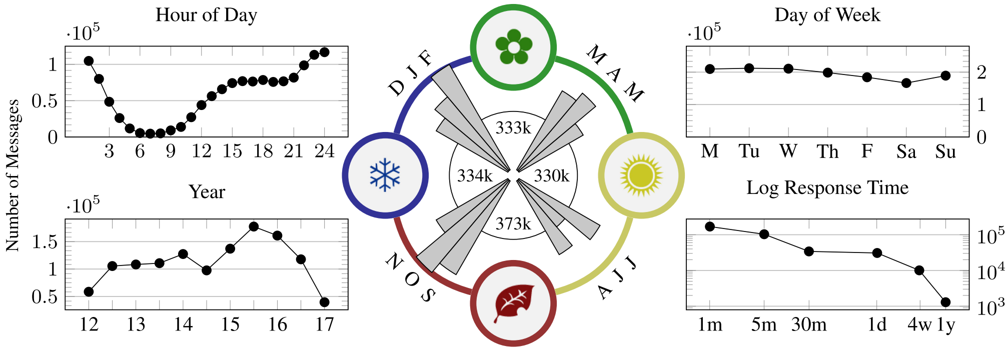

Figure 3. Distribution of messages over time. Months are grouped by season. Totals per season are listed in the inner circles with bars from 85k to 115k messages.

Table 2. Two examples of five-message context windows (c_1 and c_2) in our dataset.

No.	Time	Message	No.	Time	Message
c_1m_0	15:45:06	P: Wanna grab coffee?	c_2m_0	12:21:00	P: Perfect!!
c_1m_1	15:45:20	A: yeah	c_2m_1	15:56:22	P: Wanna go to get Thai?
c_1m_2	15:45:25	P: Sweet!!!!	c_2m_2	16:01:18	P: I'll take it you're sleeping
c_1m_3	15:45:29	P: Meet in the lobby?	c_2m_3	16:19:59	A: Yeah
c_1m_4	15:45:52	A: okay	c_2m_4	16:20:08	A: I mean I was sleeping

Message Production Across Time

To explore messaging behavior over time, we analyze message exchange trends during conversations based on the time they were sent and speaker response time.

Figure 3 presents the distribution of messages over various periods of time: hour of the day, day of the week, season of the year, and across years. Looking at the distribution over months and seasons (middle circle), there is a slight increase during autumn. Looking at the distribution over hour of the day (top left graph), there is an increase until midnight and then a dip in the morning. Looking at the distribution over days of the week (top right graph), there is a decrease as the weekend approaches. This may be instant messaging complementing real-life communication, picking up when real-life communication slows down (beginning of the week) and dropping down when real-life communication picks up (end of the week). Finally, looking at the distribution across years (bottom left graph), there is a peak in late 2015, which might be related to life events, such as starting a new job or starting school.

Figure 3 also shows the distribution of message response times with a log–log scale (bottom right). The graph shows that usually responses occur within a half-hour interval, though there are many up to a day apart, and some a year or more apart.

PREDICTING CONVERSATIONAL ASPECTS

We consider two prediction tasks related to conversation: 1) predicting the next message in a conversation, and 2) predicting message response times. Our experiments are conducted on context-windows consisting of one message written by a participant and the four preceding messages. Table 2 shows two examples of context-windows.

Features

Our features are inspired by the group and message production analysis above as well as linguistic aspects in conversational analysis. Personality of the speaker would be a relevant feature but is not feasible for us to obtain ground truth as it would require each speaker to take a personality test. Future work could attempt to

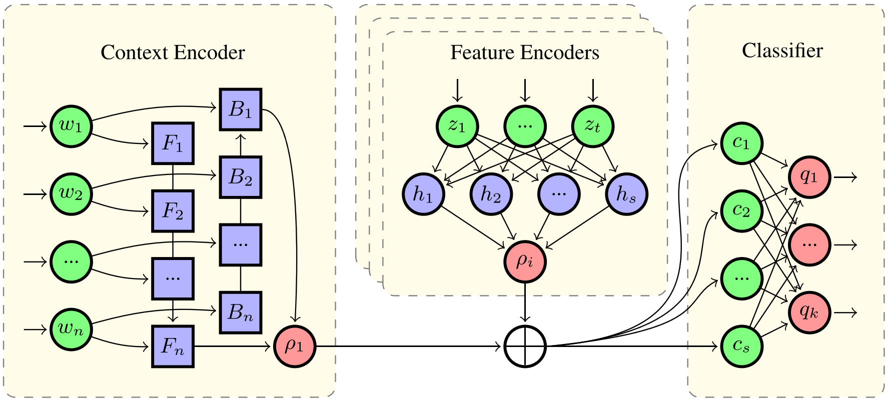

Figure 4. Model architecture encodes a context window as a sequence of tokens. The encoding is used with our other feature sets for classification.

gather this data or use a pretrained model for extracting personality from documents.[16] We define several linguistic, time, frequency, and interaction features:

Speaker Attributes: These features aim to represent the relationship(s) between the participant and their conversation partner. We derive binary features representing the seven attributes listed in Section “DATASET” for the current conversation partner. If all messages in the context window belong to the participant this vector contains only zeros.

Messaging frequency: This set of features attempts to capture the message frequency patterns observed in Figure 3. Our features include the number of messages exchanged between conversation participants in the past day, week, month, and from all time. We also include binary features representing the sequence of conversation turns in the context window.

Message timing: During our analysis, we observed important differences in message timing across the day of the week, month, season, and year. To capture these, we define a set of features including the time elapsed during the first four messages in the context window, the number of seconds between each of the first four messages, and the day, month, year, season (winter, fall, summer, spring), and hour of the day of the fourth message.

LIWC: To capture the semantic categories of text, we use the Linguistic Inquire and Word Count (LIWC) lexicon. For each speaker, we calculated normalized counts for the 73 categories and use them as features along with their cosine similarity, and vector sum.

Style Matching: To incorporate information about how the interaction between the participant and their conversation partners changes over time, we calculated the degree to which the speakers match each others language. We use the Linguistic Style Matching (LSM) metric,[17] which quantifies to what extent one speaker’s language matches the language of another using eight linguistic markers from the LIWC dictionary.[18] Specifically, we calculate LSM over the last hundred messages exchanged and the difference in LSM from the beginning to the end of the context window.

Message Embeddings: We also obtain word vector representations for each message using the GloVe Common Crawl pretrained model.[19] We chose this word embedding over other off-the-shelf options because the Common Crawl data more closely resembles our data.

Model

Figure 4 presents the model graphically. We use a bidirectional long-short term memory network (BiLSTM) to encode the messages.[20] GloVe word embeddings are used as input. To encode other feature sets, we use another fully connected layer whose output is concatenated with the LSTM output. Finally, the concatenated output is passed through a projection layer to get scores over the classes. Hyperparameters for the network, including hidden layer sizes, learning rate, and number of epochs, were tuned on a validation set. We use 80% of the data for training, and 10% for validation and testing, respectively.

For each participant, we sample random contexts for training and testing. A separate personalized model is trained for each participant and evaluated on the same participant's test data. For comparison with our personalized models, we also train and evaluate models trained on general data. For each participant, the data for the general model is sampled randomly from all other participants' data. For both the general and personalized models, the test data are the same. This allows us to measure the impact of having a user-specific model.

Prediction of Next Message in the Conversation

In this task, we must predict which of a small set of messages will occur next in a conversation. This is similar to services like Google's Smart Reply (https://allo.google.com/), which suggests potential responses to email and text messages.[21] We structure the task as a multilabel classification problem. We use the top five most frequent utterances sent by each participant as classes. The classes vary slightly but typically include values like "yes," "haha," "okay," "oh," and "nice." We also include an additional category "other," which is a random sample of 1% of the messages sent by the participant (other than the most common five).

During feature extraction, we take the last message in the context window as the label to be predicted and use the previous four messages to generate features as described above. For instance, for the first example in Table 2, we assign the label "okay," as it appears in the most common set, but for the second example we assign the label "other," as this message is not one of the five most frequent messages.

Prediction of Message Response Time

In this task, we predict the time till the next message. This kind of information can be used to make conversational agents, such as Microsoft's XiaoIce, feel more natural (https://blogs.microsoft.com/ai/xiaoice-full-duplex/). We address this task as a four-class classification problem, where messages are categorized based on their response time as: 1) the response occurs within 90 s of the timestamp of the previous message; 2) between 90 s and 10 min; 3) more than 10 min but less than a day; and 4) longer than a day. For this task, the fifth message in the context window is used to determine the label, and the previous four messages are used to generate features. For example, the response time labels for the context windows shown in Table 2 are determined by the time elapsed between msg3 and msg4, which fall into the first category, i.e., the response occurred in under 90 s.

The total number of utterances per person (P +O) ranges from 15,000 to 336,000. Two people had too few common utterances for the common utterance prediction task and were excluded from these experiments. Perhaps not surprisingly, we notice that there is a large overlap in common utterances across speakers. The utterance "yes" is in the top two most frequent utterances for all speakers, and laughter ("haha") appears in the top two in six of the eight participants. We also consider general and personalized models for this task, with data prepared in the same way as in the message prediction task.

Table 3. Prediction results averaged across participants.

	Next Msg.	Resp. Time
Majority Class	32.5	65.8
General MEmb	38.0	68.0
General All Feat	39.0	70.5
Personal MEmb	45.5	69.6
Personal All Feat	48.3	73.4

The majority baseline is compared to models that use embeddings only and a model which uses all features under a general and personal training setting.

Results

The results for both prediction tasks are shown in Table 3. We use an average of 9500 context windows for next message prediction and 88 000 for response time. The results show that across the participants in our study, our neural model with all features and personal data performs best, improving over the classifiers that use only message embeddings or classifiers that do not use personal data (the next message task excludes two participants who had too few messages.)

Table 4. Ablation results shown for each feature type and compared to a model that uses all features, as well as baselines obtained using the majority class or message embeddings (MEmb) only.

	Next Msg.	Resp. Time
Majority Class	34.0	61.9
MEmb	46.5	64.4
MEmb + Time	47.0	67.5
MEmb + LIWC	47.6	64.4
MEmb + Style	46.7	64.4
MEmb + Freq	47.6	64.8
MEmb + Attributes	46.9	64.5
All Features	50.0	68.0

In follow up analysis, we found that as the number of messages in an individual's dataset increased, the percentage that were short also increased. These messages tend to be fast and close together, leaving less room for improvement on the response time task. Future work could explore the relationship between the number of messages in an individual's dataset and the accuracy of models trained on their data.

We also perform an ablation using data from the participant with the largest number of messages. Table 4 shows the results. For the next message prediction task, the *time*, *LIWC*, and *frequency* features give the largest improvement, increasing classification accuracy by 3.5% over the baseline message embeddings model. For response time predictions, the previous response times are the most useful feature. However, we find that the combined features give an improvement of 3.6%, or a 10% error reduction. The next most useful features are the speaker attributes and the frequency of past communication.

CONCLUSION

In this article, we studied a corpus of personal conversations consisting of the instant messaging history from eight individuals. The analysis were conducted over 1.3 million messages written over a five-year time span.

We developed several linguistic features inspired by conversational and interaction behaviors we observed in the longitudinal data. Our features include message content, style matching, time features, and speaker attributes. These features were used to address two classification tasks: predicting common messages and message response times. While the most common utterances and distribution of response times vary across speakers, we found that a classifier that relies on a combination of all proposed features and uses personal data leads to error reductions of up to 15% compared to classifiers that exclusively rely on message content or are trained on messages randomly selected from other speakers in the corpus.

Our code is publicly available (https://github.com/cfwelch/longitudinal_dialog) so that others may perform similar analysis and experiments on their own personal longitudinal data or other data, to discover patterns in messaging behavior and train models for dialog prediction tasks.

ACKNOWLEDGMENTS

This work was supported in part by the Michigan Institute for Data Science, in part by the National Science Foundation under Grant #1815291, in part by the John Templeton Foundation under Grant #61156, in part by IBM as part of the Sapphire Project, and in part by DARPA under Grant #HR001117S0026-AIDA-FP-045). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Michigan Institute for Data Science, the National Science Foundation, the John Templeton Foundation, IBM, or DARPA.

REFERENCES

1. F. Benevenuto *et al.*, "Characterizing user behavior in online social networks," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, 2009, pp. 49–62.
2. A. Java *et al.*, "Why we twitter: An analysis of a microblogging community," in *Proc. 1st Int. Workshop Social Netw. Anal.*, 2009, pp. 118–138.
3. J. Leskovec and E. Horvitz, "Planetary-scale views on a large instant-messaging network," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 915–924.
4. C. Danescu-Niculescu-Mizil and L. Lee, "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in

dialogs," in *Proc. 2nd Workshop Cognitive Modeling Comput. Linguistics*, 2011, pp. 76–87.

5. C. Danescu-Niculescu-Mizil *et al.*, "No country for old members: User lifecycle and linguistic change in online communities," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 307–318.
6. E. Forsyth *et al.*, "Lexical and discourse analysis of online chat dialog," in *Proc. Int. Conf. Semantic Comput.*, 2007, pp. 19–26.
7. E. Ivanovic, "Dialogue act tagging for instant messaging chat sessions," in *Proc. ACL Student Res. Workshop*, 2005, pp. 79–84.
8. N. Majumder *et al.*, "Dialogue RNN: An attentive RNN for emotion detection in conversations," *Proc. 33rd Conf. Artif. Intell.*, 2019.
9. C. Welch *et al.*, "Look who's talking: Inferring speaker attributes from personal longitudinal dialog," in *Proc. 20th Int. Conf. Comput. Linguistics Intell. Text Process.*, La Rochelle, France, 2019.
10. T. Holmer, "Discourse structure analysis of chat communication," *Language@ Internet*, vol. 5, 2008, Art. no. 10.
11. V. H. Tuulos and H. Tirri, "Combining topic models and social networks for chat data mining," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2004, pp. 206–213.
12. A. Sordoni *et al.*, "A neural network approach to context-sensitive generation of conversational responses," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics*, 2015, pp. 196–205.
13. R. Lowe *et al.*, "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," *Proc. SIGDIAL Conf.*, 2015, pp. 285–294.
14. K. Sun *et al.*, "DREAM: A challenge dataset and models for dialogue-based reading comprehension," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 217–231, 2019.
15. T. Chen and M.-Y. Kan, "Creating a live, public short message service corpus: The NUS SMS corpus," *Lang. Resour. Eval.*, vol. 47, no. 2, pp. 299–335, 2013.
16. N. Majumder *et al.*, "Deep learning-based document modeling for personality detection from text," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 74–79, Mar./Apr. 2017.
17. A. L. Gonzales *et al.*, "Language style matching as a predictor of social dynamics in small groups," *Commun. Res.*, vol. 37, pp. 3–19, 2009.
18. Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *J. Lang. Soc. Psychol.*, vol. 29, no. 1, pp. 24–54, 2010.
19. J. Pennington *et al.*, "Glove: Global vectors for word representation," in *Proc. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
20. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
21. A. Kannan *et al.*, "Smart reply: Automated response suggestion for email," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2016, pp. 955–964.

Charles Welch is currently working toward the Ph.D. degree at the University of Michigan. His research interests include longitudinal dialog and personalization. He is the corresponding author. Contact him at cfwelch@umich.edu.

Verónica Pérez-Rosas is an assistant research scientist at the University of Michigan. Her research interests lie in natural language processing, computational linguistics, and machine learning. Contact her at vrncapr@umich.edu.

Jonathan K. Kummerfeld is a postdoctoral researcher at the University of Michigan, working on natural language processing. Contact him at jkummerf@umich.edu.

Rada Mihalcea is a professor at the University of Michigan, working on natural language processing. Contact her at mihalcea@umich.edu.