Empathetic Dialog Systems

Pascale Fung, Dario Bertero, Peng Xu
Ji Ho Park, Chien-Sheng Wu, Andrea Madotto
Human Language Technology Center
Centre for Artificial intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology

Abstract

In this paper, we outline an approach of end-to-end interactive systems with emotional embeddings, which are transfered from a large corpus. We show how to apply emotional embeddings trained from Twitter databases with hashtags and emojis as labels in a regression task. We also show that task-oriented dialog systems can be cast in an end-to-end framework using recurrent entity networks and dynamic query memory networks. In addition, we propose to include emotional embeddings into this framework for a more empathetic human-machine interactions. Finally, we show how to train an end-to-end open-domain dialog systems with deep reinforcement learning that learns a sense of humour from TV sitcoms.

Keywords: Empathetic Machines, Emotion Embedding, Representation Learning, End-to-End Dialog Systems

1. Introduction

Research on dialog systems dated back to the Darpa Communicators project in the early 90s(Walker et al., 2001). The first application of such systems was ATIS, where the system interacts with users with mixed initiatives - questions and answers from both sides, in order to complete the task of booking airplane tickets in US cities. Ever since, the general paradigm of task-oriented dialog systems follows a close-loop of automatic speech recognition, semantic decoding, dialog management, response generation and text-to-speech synthesis. Among these, the dialog manager controls the response and action of the system given user input according to some dialog policy(Williams and Young, 2007). Dialog policies can be manually written rules. Indeed most of commercial chatbots and virtual agents today still use a pre-programmed rule-based dialog policy to control system response. Dialog policies can be probabilistic - trained from collected data samples of system-to-human or operator-to-user interactions in, say, call centres. Another class of dialog systems is what is considered as chitchat bots. Chatbots typically orient towards keeping the user interacting with the system for as long as possible, with no other task completion objective. Again, a set of dialog policies can be written to control how the chatbot response to the user. However, such chatbots are often single-turn Q&A systems(Riloff and Thelen, 2000) without considering the context or discourse of the conversation.

In this paper/talk, we propose that dialog systems, both task-oriented and chatbots, can benefit from a new paradigm of empathetic human-machine interactions. We also propose that, instead of manually written rules, the empathy module can be learned in a neural network framework. In addition, we suggest that the emotional, as opposed to cognitive, content, in natural language can be modeled in a new model of emotional embeddings, to help with natural language understanding in general, and for affect recognition in particular.

2. Methodology

In the next sections, we describe the emotional components and two end-to-end dialog systems. The former includes emotional embeddings, which is a learned word embeddings that can capture not only word semantics but also word emotion. In addition, we further train our sentiment and emotion analysis modules based on emotional embeddings, which achieved promising results in Semeval-2018 Task 1: Affect in Tweets (AIT-2018). On the other hand, we also build our end-to-end dialog systems which has the potential to further leverage our emotion components, including open domain chat-bots for humor generation using reinforcement learning, and task-oriented dialog systems using recurrent entity networks and dynamic query memory networks.

3. The Empathetic Module

3.1. Emotional embedding

Emotion is very important in human-to-human communication because humans, have evolved to express and perceive emotion in natural language, developing a sense of empathy that often bonds us together socially. For example, when someone says "I have been eating alone for three days," he/she is not merely saying that one has eaten food for three days, but is implying and conveying a message of loneliness. Such information is called emotional semantics. Hence, to achieve a better human-machine interaction, machines need to both understand emotions and be empathetic through different modalities. Many researchers have experimented on learning representations of emotions in facial, vocal, and gestural expressions (Gunes et al., 2011), but not enough exploration has yet been done in texts.

Recent representation learning works have been focusing on learning word embeddings, which embed syntactic and semantic information of words into fixed-sized vectors (Mikolov et al., 2013) (Pennington et al., 2014) based on the distributional hypothesis, and have proven to be useful in many natural language tasks (Collobert et al.,

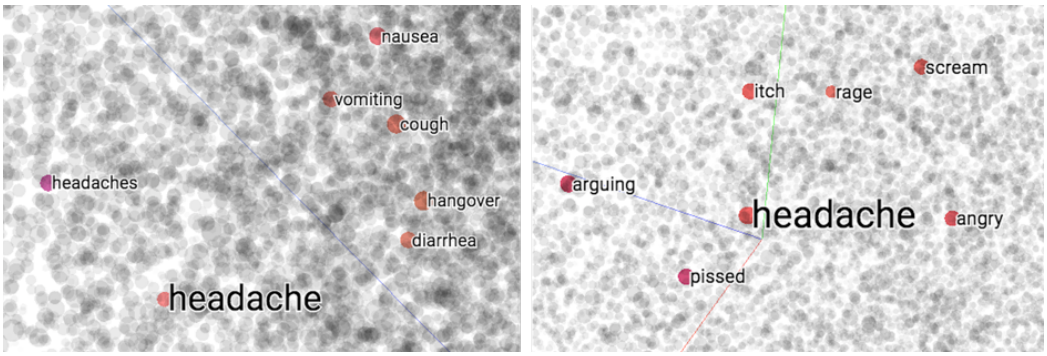

Figure 1: Visualization of word *headache* and its neighbors, left one (a) is Glove vectors and right one.

2011). However, despite the rising popularity of word embeddings, they often fail to capture the emotional valency the words convey. To elaborate, we visualized the 100-dimensional Glove vector (Pennington et al., 2014) of the word "headache" and its top 20 neighbors using cosine similarity measurement. We take the pretrained Glove vectors [1] from a 6 billion tokens corpus. The part (a) in Figure 1 shows that the Glove vector captures the semantic meaning of "headache", as shown from its neighbors like "vomiting" and "cough", but misses the emotional association that the word carries. The word "headache" in the sentence "You'll give me a headache" does not really mean that the speaker will get a headache but instead implies the anger of the speaker. Thus, we propose emotional embeddings that encode emotional semantics into fixed-sized, real-valued vectors for each word.

The most widely used word embeddings (Mikolov et al., 2013) represent each word according to its context by training from a neighboring word prediction task on a huge corpus of unlabeled data. Glove vectors use a matrix to capture the global co-occurrence statistics of words and its context. These distributed word embeddings effectively encode syntactic and semantic information of the words. Sentiment-specific word embeddings (Tang et al., 2016) encode both positive/negative sentiment and retain contextual information in a vector space. This work demonstrates the effectiveness of incorporating sentiment labels in a word-level information for sentiment-related tasks compared to existing word embeddings mentioned above. This work has led to further studies such as using document labels (Ren et al., 2016).

We trained our emotional embeddings using Convolutional Neural Network and we projected the trained vectors onto 3D space using Principal Component Analysis (PCA) and visualized each word together with the top neighboring words. We have shown that the Glove vector of indirect word "headache" cannot capture the emotion of anger inside the word "headache". We visualized the emotional embeddings of "headache" in part (b) of Figure 1, where "headache" is surrounded by such words as "pissed", "anger", "rage", and "scream" that have strong associations with anger.

[1] http://nlp.stanford.edu/data/glove.6B.zip

3.2. Sentiment and Emotion Analysis Transferring Learning

Affect in Tweets (AIT-2018) encourages more efforts in this area with the task of emotion and sentiment analysis, which is one of the most practical applications of modeling emotional text representations. We have participated in five subtasks regarding English tweets: emotion intensity regression, emotion intensity ordinal classification, valence (sentiment) regression, valence ordinal classification, and emotion classification (More details on the tasks in (Mohammad et al., 2018)).

Although these five tasks are in different formats, the most important objective is finding a good representation of the tweets regarding emotions. However, the given competition training datasets are too small to achieve our goal. Therefore, we explore utilizing larger datasets that are distantly supervised by emojis and hashtags to learn a robust representation and transfer the knowledge of each dataset to the competition datasets to solve the tasks. We aim to minimize the use of lexicons and linguistic features by replacing them with continuous vector representations.

We used external datasets, which were much larger than the competition dataset but distantly labeled with emojis (Felbo et al., 2017) and #hashtags (Wang et al., 2012), to exploit the transferred knowledge to build a more robust machine learning system to solve the task. We avoided using traditional NLP features like linguistic features and emotion/sentiment lexicons by substituting them with continuous vector representations learned from huge corpora.

We compare two models using two different emoji dataset to transform the competition data into robust sentence representations.

First model is the pre-trained DeepMoji model (Felbo et al., 2017), which is trained through emoji predictions on a dataset of 1.2 billion tweets with 64 common emoji labels. We use the pretrained deep learning network, which consists of Bidirectional Long Short Term Memory (Bi-LSTM) with attention, except the last softmax layer, as a feature extractor of the original competition datasets. As a result, each sample is transformed into a 2304-dimensional vector from the model.

The second model is our proposed emoji cluster model. We crawled 8.1 million tweets with each of which has 34 different facial and hand emojis, assuming these kinds of emojis

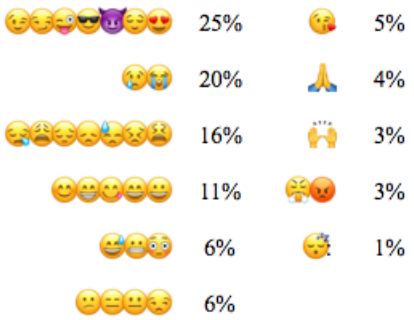

Figure 2: 11 clusters of emojis used as categorical labels and their distributions in the training set. Because some emojis appear much less frequently than others, we group the 34 emojis into 11 clusters according to the distance on the correlation matrix of the hierarchical clustering from DeepMoji and use them as categorical labels

One thing i dislike is laggers man
I hate inconsistency
The paper is irritating me
As of right now i hate dre
im sick of crying im tired of trying
why body pain why
uuugh i really have nothing to do right now
i dont wanna go back to mex
looking forward to holiday
well today am on lake garda enjoying the life
perfect time to read book
im feeling great enjoying my holiday

Table 1: Test samples from the Emoji Cluster model and their top-3 nearest sentences according to the learned representations. It shows that emotionally similar sentences are clustered together

are more relevant to emotions. Since some emojis appear much less frequently than others, we cluster the 34 emojis into 11 clusters (Figure 2) according to the distance on the correlation matrix of the hierarchical clustering from (Felbo et al., 2017). Samples with emojis in the same cluster are assigned the same categorical label for prediction. Samples with multiple emojis are duplicated in the training set, whereas in the dev and test set we only use samples with one emoji to avoid confusion. We then train a one-layer Bi-LSTM classifier with 512 hidden units to predict the emoji cluster of each sample. We take part of the dataset to construct a balanced dev set with 15,000 samples per class (total 165,000) for hyperparameter tuning and early stopping. We use 200 dimension Glove vectors pre-trained on a much larger Twitter corpus to initialize the embedding layer.

The motivation for exploring two different models is that, firstly, we want to replicate the effectiveness of using emoji for representing emotions from the previous work (Felbo et al., 2017) with a smaller dataset and a simpler model. Note that the dataset size of the emoji cluster model is less than 1% of that of the first model, whereas DeepMoji uses more than 1 billion training samples. Moreover, the first model implements a two-layer Bi-LSTM with self-attention, which has much more parameters than the second model's simple one-layer Bi-LSTM does. Secondly, we want to verify that ensembling both emoji representations trained from different datasets to boost our performance.

As a result, the model can achieve 29.8% top-1 accuracy and 61.0% top-3 accuracy on the emoji cluster prediction task. Since the objective of this model is not to predict the cluster label but to find a good sentence representation, we visualize the test set samples to discover that samples with similar semantics and emotions are grouped together (Table 1). Finally, similar to the first model, we use this model as a feature extractor on the competition datasets. Each text sample in the competition datasets is transformed into a 512-dimensional vector through the model except the last class predicting softmax layer.

We performed multiple experiments to show that emoji sentence representations and emotional word vectors trained from neural networks can be used together with tweet-specific features as input for other traditional regression models, such as SVR and Kernel Regression, to solve the task of regression and ordinal classification. We proved the effectiveness of finding the mapping of the relationship between regression and ordinal labels from the training set to perform ordinal classification. Moreover, we tried using classifier chain and regularized logistic regression methods to deal with multi-label classification.

As a final official result, our system ranked among the top three in every subtask of the competition we participated. For future work, we want to work further on employing these emotion representations on other tasks, such as text generation, while we gather more data and improve the model to train the representations.

Subtask	**System**	**Score(rank)**
1a EI-reg	SeerNet	.799(1)
	NTUA-SLP	.776(2)
	PlusEmo2Vec	.766(3)
	psyML	.765(4)
2a EI-oc	SeerNet	.695(1)
	PlusEmo2Vec	.659(2)
	psyML	.653(3)
3a V-reg	SeerNet	.873(1)
	TCS Research	.861(2)
	PlusEmo2Vec	.860(3)
	NTUA-SLP	.851(4)
4a V-oc	SeerNet	.836(1)
	PlusEmo2Vec	.833(2)
	Amobee	.813(3)
5a E-c	NTUA-SLP	.588(1)
	TCS Research	.582(2)
	PlusEmo2Vec	.576(3)
	psyML	.574(4)

Table 2: Official final scoreboard on all 5 subtasks that we participated. Scores for Subtask 1-4 are macro-average of the Pearson scores of 4 emotion categories and 5 is Jaccard index. About 35 participants are in each task.

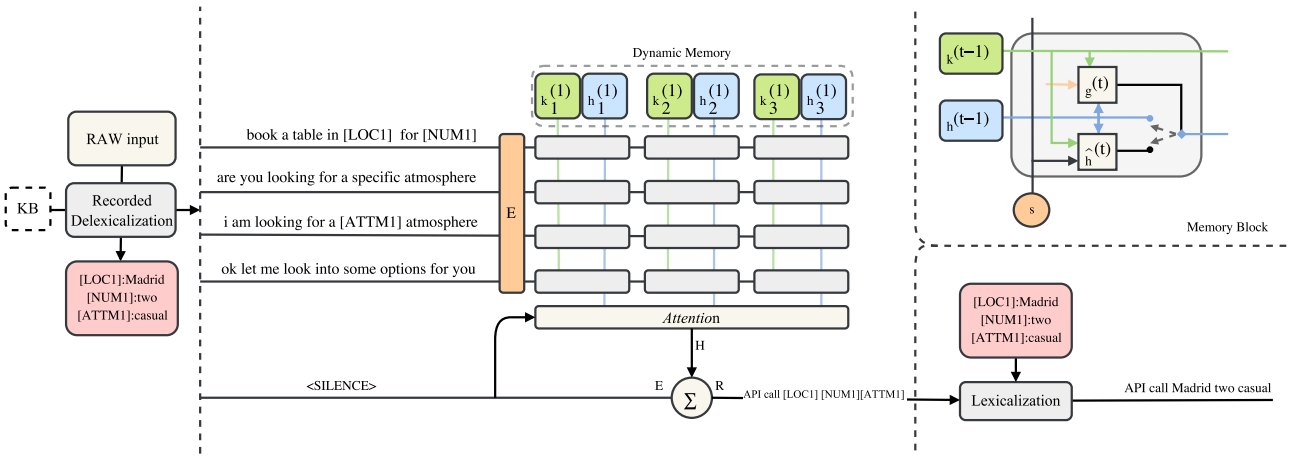

Figure 3: Recurrent entity network for task-oriented dialog systems

4. End-to-End Dialog Systems

4.1. End-to-End Task-Oriented Dialog Systems

Goal-oriented dialog requires skills such as understanding user request, asking for clarification, properly issuing API calls, querying knowledge base (KB) and interpreting query results. Traditionally, these dialog systems have been built as a pipeline, with modules for language understanding, state tracking, action selection, and language generation (Lemon et al., 2006)(Wang and Lemon, 2013)(Williams and Young, 2007). Even though those systems are known to be stable via combining domain-specific knowledge and slot-filling technique, they have limited ability to generalize into new domains and the dependencies between modules are quite complex.

End-to-end approaches train model directly on text transcripts of dialogs, and learn a distributed vector representation of the dialog state automatically (Serban et al., 2016)(Williams et al., 2017)(Zhao et al., 2017). In this way, models make no assumption on dialog state structure, holding the advantage of easily scaling up. Specifically, using recurrent neural networks (RNNs) is an attractive solution, where the latent memory of an RNN represents the dialog state. Several RNN structures have been proposed to overcome the problem (Sukhbaatar et al., 2015)(Seo et al., 2017)(Henaff et al., 2017)(Bordes and Weston, 2017)(Sukhbaatar et al., 2015), where the models are designed to represent long-term memories through global memory cells or gated functions. Here we introduce two different end-to-end models for task-oriented dialog systems, recurrent entity networks and dynamic query memory networks, respectively.

Recurrent Entity Networks Recurrent Entity Networks (REN) (Henaff et al., 2017) with dynamic long-term memory blocks have been demonstrated to have promising performance on reasoning and language understanding, which are also essential abilities for goal-oriented dialog learning. We introduced a practical task-oriented dialog framework based on Recurrent Entity Networks (REN) (Henaff et al., 2017), as shown in Figure 3. The framework is able to abstract linguistic entity by using a delexicalization mechanism. It decreases the learning complexity and outputs the next dialog utterance by choosing among action templates. The last step, lexicalization, simply replace delexicalized elements in action template with plain text based on a lookup table. Our results (Wu et al., 2017) in Dialog System Technology Challenges 6 (DSTC6) (Bordes and Weston, 2017) show that REN can achieve promising task successful rate without much hand-crafted rules. We analysed 5 incremental different settings REN: using just RDL, using temporal and user information (INFO), adding the post-processing step (POST), adding dummy user utterances in speech act (DUMMY), and QDREN model (Madotto and Attardi, 2017), which has the same architecture as REN but uses the last sentence in the memorization process. The best setting achieved an average test Precision 1 of 96.56% among all the 20 test sets evaluation. The framework is shown in Fig.3. Our code is available here [2].

Dynamic Query Memory Networks In task-oriented dialog systems, promising results have also been shown by using End-to-End Memory Network (MemNN) (Bordes and Weston, 2017)(Sukhbaatar et al., 2015)(Perez and Liu, 2017), which are neural networks with a recurrent attention model over an external memory. Besides, the multiple hop mechanism over the global memory is experimentally crucial for good performance on reasoning tasks. However, one major drawback of MemNN is that they are insensitive to represent temporal dependencies between memories.

Therefore, in Figure 4, we propose an end-to-end Dynamic Query Memory Network (DQMemNN) (Wu et al., 2018) for task-oriented dialog systems, which can be viewed as an extension of the original MemNNs. To capture the sequential dependencies of dialog utterances, we adopt the idea from (Henaff et al., 2017), whose model can be seen as a bank of gated RNNs. Therefore, DQMemNN adds a recurrent architecture between hops to obtain a similar behavior. The added dynamic component, which is a Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), is based on the utterances order appearing in the dialog history. It enables MemNN to capture the dialog's sequential dependencies by using a context-based query. Experiments show that DQMemNN outperforms original

[2] https://github.com/jasonwu0731/RecurrentEntityNetwork

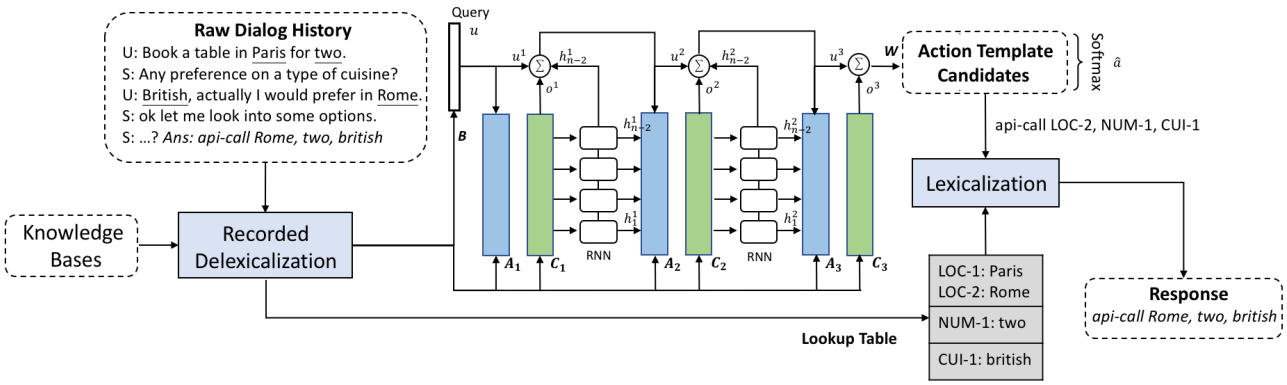

Figure 4: Dynamic query memory networks for task-oriented dialog systems

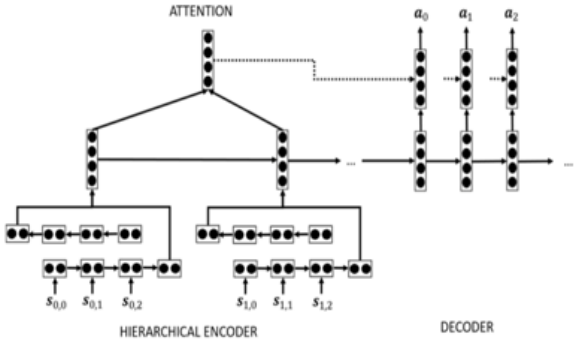

Figure 5: Seq2Seq with attention.

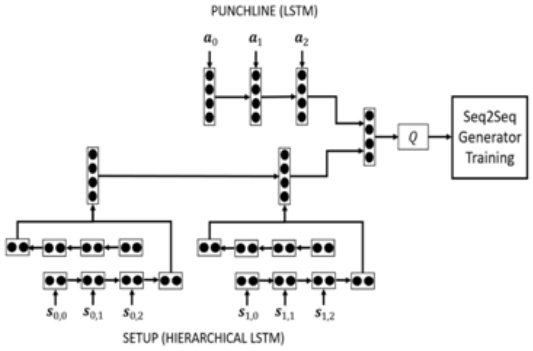

Figure 6: Seq2Seq + reinforce.

end-to-end memory network models on bAbI full-dialog task (Bordes and Weston, 2017) by 3.1% per-response and 39.3% per-dialog accuracy.

4.1.1. Incorporating Emotional Embeddings

We previously showed that it is possible to embed emotional responses by adding a specialized module capable of capturing emotions (Yang et al., 2017; Winata et al., 2017; Siddique et al., 2017). This module used an emotion classifier to condition the system answer. In end-to-end system adding such module is not trivial and we believe further study is required, especially if we want to leverage the benefit of an end-to-end system. A natural solution is to add embedding emotions in our end-to-end task-oriented dialog systems (i.e. DQMemNN and RDL+REN) models. From an high level prospective, it would be to load in memory the learned emotional representation, instead of standard embeddings. This can be implemented in many different ways: concatenation and dual memory.

Using DQMemNN as the running example[3]. The first way to implement is to concatenate the emotional embeddings with the learned ones. Practically, this is done by concatenating the standard (learned) word embeddings with the emotional embeddings presented before. In this way the model is able to retrieve words in memory based on their emotional semantics. The second way is to create two separated memory, one for the standard embeddings and one for the emotional ones. In this way the emotion is captured by a separated memory module, which would helps to generate a more emotional related response. Indeed the model is still trained end-to-end but each memory would focus in two different aspects (or prospective) of the input.

4.1.2. A Chat-Bot with a Sense of Humor

In this section, we will show how to train a chatbot to have a sense of humor in the end-to-end framework. Conversational humor is believed to be generated through two phases: "setup" and "punchline" (Attardo, 1997) (Taylor and Mazlack, 2005). In the setup, the contextual information and the setting of the jokes are introduced, and the audience is prepared to receive the humorous stimuli. It is followed by the punchline, which releases the tension built during the setup phase to trigger a reaction, usually laughter.

We propose a reinforcement learning framework based on sequence-to-sequence LSTM language model (Sutskever et al., 2014) (Serban et al., 2016) (Ranzato et al., 2016) to generate humorous punchlines for a given setup context.

Our work aims to address two main issues with current state-of-the-art humor generation systems. The first one is the dependence on a limited number of fixed, hand-crafted templates. Using repetitive joke structures may bore the

[3]It would work also for RDL+REN

audience. Other than in response to the popular "tell me a joke" request these methods may seldom be used in other contexts. This also introduces the second issue, which is the lack of context of the joke produced. Most of the current humor generation systems (Binsted and Ritchie, 1994) (Hossain et al., 2017) just produce a large amount of generic and isolated jokes.

End-to-end dialog generation models (Serban et al., 2016) (Li et al., 2016) were proposed a ways to generate relevant utterances in response to a given input context. While the sentences generated by these methods generally match the discourse context, they are not specifically designed to generate funny jokes. Even if sometimes they manage to produce a funny response, it is mostly a random side-effect or it is due to grammar and syntax mistakes in the generation rather than due to a well-built punchline. To overcome the shortcomings of both template-based and end-to-end methods, we are interested in using the contextual information provided in the setup prompts, to generate appropriate punchlines that are both funny and relevant to the conversation. We use the sitcom canned laughter as labels to train the funniness reward (Bertero and Fung, 2016) (Chen and Lee, 2017), and the distinction between reference sentences (always relevant) and generated ones (not always relevant) to train the relevance reward.

We train a hierarchical sequence-to-sequence model in order to maximize a reward score based on funniness and relevance, and obtain more funny punchlines that are coherent to the setup sentences.

5. Conclusion

In this paper, we have outlined an approach of end-to-end interactive systems with emotional embeddings. We showed how to apply emotional embeddings trained from Twitter databases with hashtags and emojis as labels in a regression task of SemEval 2018 task. In this latter, our system ranked among the top three in each subtask. We also showed that task-oriented dialog systems can be cast in an end-to-end framework using recurrent entity networks and dynamic query memory networks, by showing top results in bAbI Dialog dataset and DSTC6. We also proposed to include emotional embeddings into this framework for a more empathetic human-machine interactions. Finally, we showed how to train an end-to-end chatbot with reinforcement deep learning that learns a sense of humour from TV sitcoms.

6. Acknowledgements

7. References

Attardo, S. (1997). The semantic foundations of cognitive theories of humor. *Humor-International Journal of Humor Research*, 10(4):395–420.

Bertero, D. and Fung, P. (2016). A long short-term memory framework for predicting humor in dialogues. In *Proceedings of NAACL*.

Binsted, K. and Ritchie, G. (1994). An implemented model of punning riddles. Technical report, University of Edinburgh, Department of Artificial Intelligence.

Bordes, A. and Weston, J. (2017). Learning end-to-end goal-oriented dialog. *ICLR*, abs/1605.07683.

Chen, L. and Lee, C. M. (2017). Convolutional neural network for humor recognition. *arXiv preprint arXiv:1702.02584*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *Conference on Empirical Methods in Natural Language Processing(EMNLP) 2017*.

Gunes, H., Schuller, B., Pantic, M., and Cowie, R. (2011). Emotion representation, analysis and synthesis in continuous space: A survey. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 827–834. IEEE.

Henaff, M., Weston, J., Szlam, A., Bordes, A., and LeCun, Y. (2017). Tracking the world state with recurrent entity networks. *ICLR*, abs/1612.03969.

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

Hossain, N., Krumm, J., Vanderwende, L., Horvitz, E., and Kautz, H. (2017). Filling the blanks (hint: plural noun) for mad libs humor. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 649–658, Copenhagen, Denmark, September. Association for Computational Linguistics.

Lemon, O., Georgila, K., Henderson, J., and Stuttle, M. (2006). An isu dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the talk in-car system. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, pages 119–122. Association for Computational Linguistics.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas, November. Association for Computational Linguistics.

Madotto, A. and Attardi, G. (2017). Question dependent recurrent entity network for question answering. *NL4AI*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mohammad, S. M., Bravo-Marquez, F., Salameh, M., and Kiritchenko, S. (2018). Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Pennington, J., Socher, R., and Manning, C. (2014).

Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perez, J. and Liu, F. (2017). Gated end-to-end memory networks. In *EACL*.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2016). Sequence level training with recurrent neural networks. *Proceedings of International Conference on Learning Representations*.

Ren, Y., Wang, R., and Ji, D. (2016). A topic-enhanced word embedding for twitter sentiment classification. *Information Sciences*, 369:188–198.

Riloff, E. and Thelen, M. (2000). A rule-based question answering system for reading comprehension tests. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems-Volume 6*, pages 13–19. Association for Computational Linguistics.

Seo, M., Min, S., Farhadi, A., and Hajishirzi, H. (2017). Query-reduction networks for question answering. *ICLR*.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.

Siddique, F. B., Kampman, O., Yang, Y., Dey, A., and Fung, P. (2017). Zara returns: Improved personality induction and adaptation by an empathetic virtual agent. *Proceedings of ACL 2017, System Demonstrations*, pages 121–126.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., and Zhou, M. (2016). Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):496–509.

Taylor, J. and Mazlack, L. (2005). Toward computational recognition of humorous intent. In *Proceedings of Cognitive Science Conference*, pages 2166–2171.

Walker, M., Aberdeen, J., Boland, J., Bratt, E., Garofolo, J., Hirschman, L., Le, A., Lee, S., Narayanan, S., Papineni, K., et al. (2001). Darpa communicator dialog travel planning systems: The june 2000 data collection. In *Seventh European Conference on Speech Communication and Technology*.

Wang, Z. and Lemon, O. (2013). A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *SIGDIAL Conference*, pages 423–432.

Wang, W., Chen, L., Thirunarayan, K., and Sheth, A. P. (2012). Harnessing twitter" big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 587–592. IEEE.

Williams, J. D. and Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Williams, J. D., Asadi, K., and Zweig, G. (2017). Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*.

Winata, G. I., Kampman, O., Yang, Y., Dey, A., and Fung, P. (2017). Nora the empathetic psychologist. *Proc. Interspeech 2017*, pages 3437–3438.

Wu, C.-S., Madotto, A., Winata, G., and Fung, P. (2017). End-to-end recurrent entity network for entity-value independent goal-oriented dialog learning. In *Dialog System Technology Challenges Workshop, DSTC6*.

Wu, C.-S., Madotto, A., Winata, G., and Fung, P. (2018). End-to-end dynamic query memory network for entity-value independent task-oriented dialog. In *ICASSP*.

Yang, Y., Ma, X., and Fung, P. (2017). Perceived emotional intelligence in virtual agents. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, pages 2255–2262, New York, NY, USA. ACM.

Zhao, T., Lu, A., Lee, K., and Eskenazi, M. (2017). Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 27–36. Association for Computational Linguistics, August.