Deep AM-FM: Toolkit For Automatic Dialogue Evaluation

Chen Zhang, Luis Fernando D'Haro, Rafael E. Banchs, Thomas Friedrichs, Haizhou Li

Abstract There have been many studies on human-machine dialogue systems. To evaluate them accurately and fairly, many resort to human grading of system outputs. Unfortunately, this is time-consuming and expensive. The study of AM-FM (Adequacy Metric - Fluency Metric) suggests an automatic evaluation metric, that achieves good performance in terms of correlation with human judgements. AM-FM framework intends to measure the quality of dialogue generation along two dimensions with the help of gold references: (1) The semantic closeness of generated response to the corresponding gold references; (2) The syntactic quality of the sentence construction. However, the original formulation of both adequacy and fluency metrics face some technical limitations. The latent semantic indexing (LSI) approach to AM modeling is not scalable to large amount of data. The bag-of-words representation of sentences fails to capture the contextual information. As for FM modeling, the n-gram language model implementation is not able to capture long-term dependency. Many deep learning approaches, such as the long short-term memory network (LSTM) or transformer-based architectures, are able to address these

Chen Zhang
National University of Singapore (NUS), 4 Engineering Drive 3 Block E4, #06-20, Singapore e-mail: e0397123@u.nus.edu

Luis Fernando D'Haro
ETSI de Telecomunicación, Universidad Politécnica de Madrid (UPM), Avenida Complutense 30, 28040 Madrid, Spain e-mail: luisfernando.dharo@upm.es

Rafael E. Banchs
School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore e-mail: rbanchs@ntu.edu.sg

Thomas Friedrichs
Robert Bosch (SEA) Pte Ltd, 11 Bishan Street 21, Singapore e-mail: Thomas.Friedrichs@sg.bosch.com

Haizhou Li
National University of Singapore (NUS), 4 Engineering Drive 3 Block E4, #06-20, Singapore e-mail: haizhou.li@nus.edu.sg

issues well by providing better contextual-aware sentence representations than the LSI approach and achieving much lower perplexity on benchmarking datasets as compared to the n-gram language model. In this paper, we propose deep AM-FM, a DNN-based implementation of the framework and demonstrate that it achieves promising improvements in both Pearson and Spearman correlation w.r.t human evaluation on the bench-marking DSTC6 End-to-end Conversation Modeling task as compared to its original implementation and other popular automatic metrics.

1 Introduction

Recently, conversational AI has become a hot research topic. With the proliferation of human-human dialog data, sophisticated learning strategies and boost in computational power, training end-to-end conversational dialogue system becomes feasible. One key step to the development of such systems is the evaluation metric. The evaluation of conversational dialogue systems is hard, because conversational dialogue evaluation is not as straight forward as having a single objective metric to optimize. What constitute high-quality dialogue is rather complex. Common practice generally involves human judges to provide ratings to system-generated responses, but it is neither cost-effective nor time-efficient. Most commonly used automatic evaluation metrics are shown to correlate poorly with human judgements [13]. The AM-FM (Adequacy Metric - Fluency Metric) framework, which is originally proposed for evaluating machine translation systems [1], has been adopted to address this problem [7].

The original implementation of adequacy metric leveraged latent semantic indexing [12], where 10K sentences from the twitter training corpus were randomly selected for training the singular value decomposition (SVD). Sentences were represented by bag-of-words features in the term-document matrix. Despite its good dialogue evaluation capability [7], this technique has serious drawbacks. Firstly, the bag-of-words representation fails to capture contextual information of words in a sentence. In addition, the sentences are randomly picked among the training corpus, this fails to account for the logical continuations between consecutive utterances in a dialogue. Moreover, when the data size increases, the term-document matrix can be very large and unable to fit into the memory. As a result, it is infeasible to perform SVD numerically.

Many studies have been devoted to learn effective word-level and sentence-level embeddings in the continuous space that are able to capture contextualized information from a large amount of text data. Recent advancements in deep learning techniques have brought promising prospects in this area. For example, [20] proposes the use of long short-term memory (LSTM) network [9] for learning sentence-level embeddings, which obtains outstanding performance in the web search task. Contextualized word embedding learnt with bidirectional LSTM [21] or transformer-based architectures [6, 22] greatly helps tackle many of the NLP benchmark tasks, such as question answering, semantic similarity and natural language inference.

The n-gram language model is used for implementing the fluency metric in the initial setup. Despite its simplicity, it provides competitive results [15]. But clearly it faces several inherent limitations. Firstly, it neglects the long distance dependencies. In the dialogue setting, long-term dependency among the user-system interaction is important for understanding the dialogue. In addition, count-based approaches neglect the ordering of words, which may be essential for understanding the context. Moreover, this approach has a structural problem since various smoothing techniques are required to account for the unseen cases. In recent years, the language modeling research has been filled with different deep learning approaches to address the long-term dependency issue of n-gram language model. LSTM-based and transformer-based approaches [15, 5, 22] have been successful in several benchmark datasets, for example, the One Billion Word dataset [3].

Hence, we are motivated to explore alternative DNN-based implementations to AM-FM framework and intend to present the initial version of the toolkit based on these implementations[1] in this paper. We compare our implementations with the original setup to demonstrate that deep-learning techniques are effective and scalable for modeling both the AM and the FM component. The organization of the paper are as follow: Section 2 discusses the background of AM-FM framework and the relevance of deep learning approaches. Section 3 shares details of implementations for the adequacy component. Section 4 focuses on the fluency component. Section 5 discusses the experimental results. The last section concludes this paper and layouts the future plan for improving the toolkit.

2 Related Work

In this section, we would like to give a brief background of AM-FM framework and motivate the deep learning approach to AM-FM.

2.1 AM-FM Framework

The AM-FM framework, originally proposed in [1], is used to evaluate machine translation systems. In a typical evaluation process, we need to assess translated sentences of different systems with respect to multiple human references and provide a score to each system for ranking purpose. Usually, there are human judges scoring the systems and the proposed automatic evaluation metric should correlate well with the human scores. AM-FM framework aims to achieve this by evaluating translations along two dimensions, the adequacy and the fluency, which are metrics designed to address independently the semantic and syntactic aspects of the translation. The semantic aspect serves to assess how much source information is

[1] https://github.com/e0397123/AM-FM-PM.git

preserved by the translation whereas the syntactic aspect evaluates the linguistic quality of the translation. A continuous space model is adopted for assessing adequacy whereas an n-gram language model is used for evaluating fluency. Both metrics operate at the sentence-level. For computing the AM score of a system response, a term-document matrix corresponding to the target language is constructed. Sentences are represented with bag-of-words features and mapped to low-dimensional continuous vectors leveraging singular value decomposition (SVD). The cosine distances between the response vector and each of the reference vectors are computed. The maximum cosine distance is retained as the AM score of the particular system response. For evaluating fluency, an n-gram language model is implemented with the target language data. Then the model is used to compute normalized log probabilities of system responses, which correspond to their respective FM scores. The AM and FM scores of a particular system response are then combined to form a final evaluation score based on different strategies, such as the harmonic mean and the geometric mean.

The dialogue evaluation process is similar to that of machine translation in the sense that user queries are equivalent to the source sentences and dialogue system responses are equivalent to the translation system responses. The quality of dialogue generation is evaluated by comparing dialogue system responses against multiple corresponding human-written references. This motivates the extension of AM-FM framework to the dialogue setting [7].The same techniques are adopted for implementing adequacy and fluency metrics with some minor modifications, such as for the FM modeling, a relative-scale scoring mechanism is introduced: $FM_{score} = \frac{min(prob_{candidate}, prob_{reference})}{max(prob_{candidate}, prob_{reference})}$, instead of using the absolute log-probability score so as to incorporate human references in FM computation. Specifically, the metric is tested on the evaluation of 20 submitted systems to `End-to-End Conversational Modeling Track` of DSTC-6 challenge [2] [10]. The test set contains 2000 dialogue contexts and 11 references per context. System responses are compared against the references and evaluated by 10 human judges. Rating at the utterance level is obtained by computing the average of ratings given by the judges to a particular system response and system-level rating is computed by averaging utterance-level ratings of all responses to the 2000 dialogue contexts. [7] demonstrates that AM-FM framework is capable of generating similar system-level ratings w.r.t the above-mentioned human ratings.

2.2 Relevance of Deep Learning

Despite AM-FM's good evaluation capability, we would like to address its current limitations leveraging deep learning, which has revolutionized many areas: computer vision, speech recognition, natural language processing, robotics, etc. We primarily discuss the application of deep learning techniques in vector-space repre-

[2] http://workshop.colips.org/dstc6/index.html

sentations of word or sentence meanings and language modeling pertaining to the AM-FM framework in this section.

Word Embedding [21] proposes ELMo, a deep contextualized word representation model, which leverages bidirectional language models. Feature representations are extracted from both the left-to-right and a right-to-left language models and concatenate together for other downstream tasks. This approach has achieved significant improvement in several NLP benchmarking tasks, such as question answering, name entity recognition and sentiment analysis. [6] marks a departure from traditional left-to-right or right-to-left language model training by adopting the masked language model objective for pretraining, where a portion of wordpiece tokenized input sequence are masked and the model is supposed to predict the masked tokens. The model is also jointly trained the next-sentence-prediction (NSP) objective to identify whether one sentence is a correct continuation of another. It is a deep bidirectional model with multilayer of transformer encoders [24]. Just like ELMo, contextualized word embeddings can be extracted from the trainined BERT model for many other NLP tasks.

Sentence Embedding For setence-level embedding, [11] proposes the Skip-Thought Vectors. The main idea is to encode the target sentence with an recurrent neural network (RNN) encoder and reconstruct the previous and next sentences with two separate RNN decoders. The final hidden state of the RNN encoder is used as an embedding for the target sentence. De-noising autoencoder approach is adopted by [8] whereby the model need to reconstruct the original sentence after it's getting some parts changed or deleted. [16] proposes quick-thoughts, an approach to predict whether a context is correct for a given sentence and a classifier is trained to differentiate context sentences from other contrasting sentences based sentences and their corresponding labeled contexts.

Language Modeling Deep learning techniques are also useful for language modeling. [19] proposes the first recurrent neural network language model (RNNLM). RNNLM performs better than the feed-forward neural network language model and RNNs are able to processing variable-length sequences. [23] brought LSTM [9] into language modeling and proposed LSTM-RNNLM to address the issue of capturing the long-term dependency. [18] proves that attention mechanism is useful for RNN-based language modeling in the context of coherent dialogue. After the invention of transformer [24], lots of transformer-based language models [6, 5, 22] have greatly impacted the NLP field.

3 Adequacy Metric

We explore the use of transfer learning for adequacy metric modeling. Recently, transfer learning has become prevalent in NLP whereby a general language model is pretrained on a large amount of text data. It is perceived that the model contains

general-purpose abilities and knowledge that can be transferred to the downstream tasks. The contextualized embeddings extracted from these models are perceived to provide meaningful representations of words or sentences, which are key to the adequacy metric.

Why BERT For general language model pretraining, the Bidirectional Encoder Representation from Transformers (BERT) [6] is chosen. The reason is that BERT consists of two optimization objectives, one of which is the `next sentence prediction (NSP)`. This is very similar to the dialogue setting in the sense that the system response should be a logical continuation to a user query. The NSP objective ensures that the model is able to capture inter-sentences relationships. For transfer learning, parameter transfer is adopted as the general model is pretrained on a different domain as compared to that of the dialogue evaluation task. In order to adapt to the target domain, we need to continue training the model with the collected twitter dialogue data. With parameters transferred from the pretrained model as an initial starting point,we can lead to faster adaptation. Fig. 1 presents an example input to the BERT pretraining pipeline.

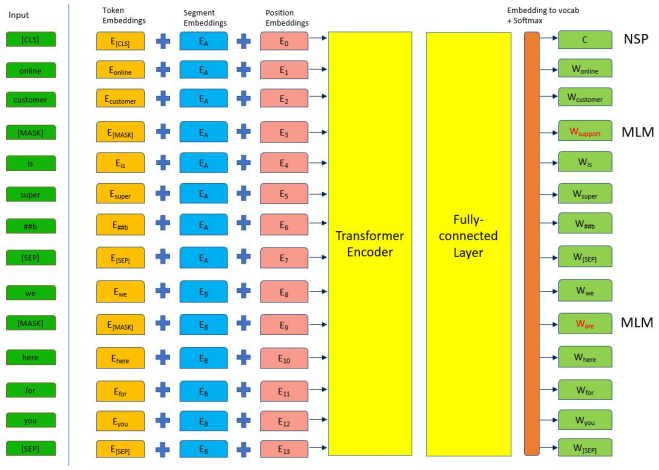

Fig. 1: Twitter Dialogue Example Input to BERT Pretraining Pipeline

Input Representation In the context of twitter dialogue, Let U_i denote the user query and R_i denote the response of the customer support. The sentences is tokenized into wordpiece tokens, $U_i = \{w_j\}_{(j=1)}^{m}$ and $R_i = \{w_j\}_{(j=1)}^{n}$. They form a pair by adding a `[CLS]` token in front, a `[SEP]` token to separate both tokenized sequences and a `[SEP]` at the end. The total sequence length is $n+m+3$ and it is a hyperparameter that can be arbitrarily set. Given that twitter sentences are generally short, a total sequence length of 60 is suitable for the experiments. For each input token, w_j, the input to the network is $[E_j^{token} + E_j^{segment} + E_j^{position}]$.

Special Tokens The `[CLS]` is a special token for classification. During training, it goes through the transformer encoder layers and the final hidden state is transformed into a 2x1 shaped vector with a classification layer. The is-next-sentence probability is calculated with softmax. The `[SEP]` token acts as the boundary of sentences. The special `[MASK]` token serves the Masked Language Model (MLM) training objective. A ratio of the whole input token sequences are masked at random and replaced with `[MASK]` during training. The ratio is kept at 0.15 across all experiments. Masked tokens are fed into the network together with the rest of the tokens. The output vectors from the fully-connected layer are multiplied with the embedding matrix and after softmax computation, a probability distribution across the entire vocabulary for each token is obtained. The cross entropy loss is adopted in the training process and the network is supposed to optimize the accuracy of predicting the correct tokens corresponding to the masked input tokens.

Embedding Extraction The parameters of the pretrained model, `[BERT-Base, Multilingual Cased]`, are transferred as an initial starting point and training is continued with the target domain twitter dialogue data. The details regarding training BERT models can be found in [6]. The trained model is then used as a feature extractor to get meaningful representations of the sentences. In all the experiments, sentence embeddings are obtained by applying heuristical operations on the extracted word-level embeddings. Let $H_{i,j}$ denotes the hypothesis of system i in response to dialogue context j and $R_{k,j}$ denotes the k-th ground-truth reference to dialogue context j. The same pre-processing steps are performed on both $H_{i,j}$ and $R_{k,j}$ and then they are fed into the trained network. The corresponding activations of the top hidden layer are extracted. The final sentence-level embeddings, $\mathbf{E}_{i,j}$ of $H_{i,j}$ are computed by averaging the embeddings of its corresponding word vectors following Eq. 1. Here, w refers to an individual token while $\mathbf{e}_w$ represents the extracted embedding of token w. The sentence embedding of $R_{k,j}$ is obtained in the same way.

$$\mathbf{E}_{i,j} = \frac{\sum_{w \in \mathbf{H}_{i,j}} \mathbf{e}_w}{|\sum_{w \in \mathbf{H}_{i,j}} \mathbf{e}_w|} \tag{1}$$

Adequacy Score Computation Given the sentence embedding mentioned in the previous paragraph, a final system-level adequacy score can be computed. The sentence embedding of each system submission, $\mathbf{E}_{i,j}$ is compared against that of each corresponding human references, $\mathbf{E}_{k,j}$. Eq. 2 shows the way to compute $AM_{i,j}$ (the utterance-level adequacy score per system). After measuring the cosine similarity between each system response and all eleven human references, the maximum score is retained. The final system-level score, AM_j, is obtained by averaging all two thousand utterance-level scores (shown in Eq. 3).

$$AM_{i,j} = max_{k \in \{1,2,\ldots,11\}} \frac{\mathbf{S}_{i,j}^T \cdot \mathbf{S}_{i,k}}{|\mathbf{S}_{i,j}||\mathbf{S}_{i,k}|} \tag{2}$$

$$AM_j = \frac{\sum_{i=1}^{2000} AM_{i,j}}{2000} \tag{3}$$

4 Fluency Metric

The key to fluency metric is a good language model (LM), which is able to approximates the true distribution of text data well. Traditional statistical LMs try to assign probabilities to a word sequence via the product rule of conditional probability (Eq. 4). The n-gram language model is based on the n-order Markov assumption, which states that the current word depends only on the previous (n-1) words. Smoothing techniques were introduced to counteract the model's problem of assigning zero probabilities to unseen n-grams.

$$P(S) = P(w_1, w_2, ..., w_m) = P(w_1)P(w_2|w_1)....P(w_m|w_1, w_2, ..., w_{(m-1)}) \quad (4)$$

Why LSTM-RNN LM Though smoothing helps, n-gram models suffer from other major issues like curse of dimensionality and inability to capture long-term information. These problems limit FM's ability to accurately assess the syntactic quality of sentences. In order to address these issues, Recurrent neural network language models (RNN-LM) [19] is explored, especially the use of long short-term memory cell [9]. The reasons why this family of language models are chosen include: 1. RNN is inherently suitable for processing variable-length sequence data in terms of its structure. 2. Especially with the LSTM cell, the network is able to retain long-term information as LSTM is proposed to address the vanishing gradient problem of vanilla RNN cell. 3. This family of network has been proven to achieve much lower perplexity on many benchmarking test sets in the literature as compared to the n-gram model.

LSTM-RNN LM Implementation LSTM-RNN LM [23] is similar to RNNLM[19] except the incorporation of LSTM cell. The objective of a RNN language model is to predict the next token based on the previous context in a left-to-right manner whereby the token at the current time step in the target sequence is predicted with a softmax layer on top of the linear transformation of the current time-step hidden state, which is dependent on the hidden state of previous time steps and the current input token. It is assumed that the hidden state of the RNN carries forward the information of all the previous time steps. The target sequence, which serves as the supervision to the model during training, is one token ahead of the input sequence. The token can be a word, a n-gram or a character. The problem can be formulated as Eq. 5 and Eq. 6 where $\hat{y_t}$ is the t_{th} prediction out of target vocabulary V and w_t denotes the target word at time step t that maximizes the conditional probability.

$$\hat{y_t} = softmax(W_{softmax}h_t + b_{softmax}) \quad (5)$$

$$w_t = argmax_{w_t \in V} P(\hat{y_t}|V, \hat{y_1}, ..., y_t \hat{-} 1) \quad (6)$$

The recurrent unit, which suffers from the problem of vanishing gradient, is replaced by a LSTM cell, which contains three gate structures (the input, output and forget gate) to control the flow of information. The details regarding LSTM can be

found in [9, 23]. Input sequences are preprocessed in the same way as that used in AM modeling. Sentencepiece tokenization [14] is performed on the input sequence to handle the out-of-vocabulary problem. The tokenizer is trained on the full training set with a total vocabulary size of 32000. Tokenized sequences are then fed into the LSTM-RNN LM for training. Word embedding is pretrained with the continous-bag-of-words algorithm and then used to initialize weights of the embedding layer. Stochastic gradient descent is chosen to be the optimizer, because in the literature, SGD has been empirically found to outperform other methods for the language modeling task [15]. A dropout of 0.2 is chosen to avoid over-fitting.

LSTM-RNN LM For Fluency Estimation The computations of utterance-level and system-level FM scores are almost the same as those mentioned in [7] except that instead of directly computing the normalized log probability of a sentence by summing up the log probabilities of all tokens and dividing by the number of tokens, the inverse of sentence-level perplexity is used. This is shown in Eq. 7 where $P(R)_{norm}$ denotes the normalized log probability of a reference sequence, R. $PP(R)$ refers to the sentence-level perplexity and N is the number of tokens in R. Eq. 8 indicates that $PP(R)$ can be obtained by averaging the sum of cross-entropy loss for predicting each token in R and then applying an exponential function on the averaged value.

$$P(R)_{norm} = exp(\frac{log(P(R))}{N}) = \frac{1}{PP(R)} \tag{7}$$

$$PP(R) = P(R)^{-\frac{1}{N}} => log(PP(R)) = \frac{1}{N} log(\frac{1}{P(R)}) \tag{8}$$

After getting the sentence-level normalized probability, the utterance-level fluency score, $FM_{i,j}$, is obtained firstly by computing the ratio of $min(P(H)_{i,j}, P(R)_{i,k})$ and $max(P(H)_{i,j}, P(R)_{i,k})$[3]. Given that there are multiple references per each test case, $FM_{i,j}$ is then calculated as the difference between the maximum ratio and the minimum ratio. This way, a small score difference indicates that the system is only able to generate an averaged response when compared to the eleven gold references; however, if the difference is large, the system is able to generate a response that is more closer to one of the valid references. Therefore, the new formulation allows a better discrimination between the different systems. Lastly, the system-level score, FM_j is computed by averaging sum of all $FM_{i,j}$.

5 Experimental Results

Experiment Setup The dialogue dataset is mainly about conversations between users and companies' customer support on Twitter. We followed the instructions

[3] Here R refers to the reference, H denotes the system response, i is the index of test case, j denotes the index of a specific system and k is the index of a reference

provided by the organizer of DSTC6 End-to-End-Conversation-Modeling Track to collect the official training set and validation set[4]. There are around 1.7 million dialogues in the training set. For AM Modeling, experiments are conducted across different sizes of training data: 10K, 20K, 50K and 100K of twitter dialogues. The validation set contains 52682 dialogues. As mentioned in Sec. 2, the test set contains 2000 dialogues, which are reserved for conducting correlation analysis between deep AM-FM and the human judgements.

Performance of AM Modeling Table. 1 presents the Masked-Language-Model (MLM) accuracy as well as Next-Sentence-Prediction (NSP) accuracy on the validation set after training BERT with different size of twitter training data. Since, BERT is a form of auto-encoding (AE) language model, a higher MLM accuracy indicates it has a stronger ability to predict the masked tokens, therefore rendering a better AE language model. Moreover, a higher NSP accuracy indicates it has a stronger ability to discriminate relevant responses to the corresponding context from the irrelevant ones. The optimization of these two objectives depends on the model's ability to capture the contextual information in the text. Model with high MLM and NSP accuracy therefore can better represent the semantics of the words or sentences. Hence, it can be concluded that better word or sentence embeddings can be learnt with the presence of more data, because with increasing amount of data, generally higher MLM and NSP accuracy are achieved. The deep learning implementation enables the leverage of the power of more data. This is in contrast to the slow computation of singular value decomposition when the data size is large and the constraint imposed by the memory size. All experiments are conducted on a single Nvidia GTX 2080-Ti GPU with the BERT implementation. The training time varies from a few hours to one day across different training data sizes.

Table 1: MLM & NSP accuracy across different training data size

Train Size[a]	MLM Accuracy	NSP Accuracy
10K	0.6170	0.8850
20K	0.6364	0.8975
50K	0.6460	0.9012
100K	0.6452	0.9100

[a] The data size corresponds to the number of dialogues.

Performance of FM Modeling The performance of the LSTM-RNN LM implementation is compared to different n-gram models (plus Kneser-Ney smoothing) across different training data sizes in terms of perplexity. The results are shown in table. 2. It can be observed that LSTM-RNN LM consistently outperforms the n-gram models across all training data sizes because for the same amount of training

[4] Refer to https://github.com/dialogtekgeek/DSTC6-End-to-End-Conversation-Modeling.git for the data collection process

data, LSTM-RNN LM is able to achieve much lower perplexity than the rest of n-gram LMs on the validation set. It can be speculated that with even larger amount of training data, LSTM-RNN LM will perform even much better. Same as the setup in AM implementation, all experiments are conducted on a single Nvidia GTX 2080-Ti GPU. The training time varies from a few hours to few days depending on the training size. Even though when the data size becomes huge, LSTM-RNN LM will take a long time to finish training. The LSTM cell can be replaced by gate recurrent unit [4], which performs similar to LSTM, but only with two gates for controlling the information flow, rendering its training process to be faster.

Table 2: Perplexity for different models on valid set[b] across different training data size

Train Size	uni-gram	bi-gram	3-gram	4-gram	5-gram	LSTM-RNN LM
10K	597.78	230.72	199.81	201.93	204.61	**122.82**
20K	635.49	227.50	191.45	193.08	196.00	**117.29**
50K	666.71	222.77	180.52	180.16	183.05	**122.90**
100K	682.99	218.59	171.45	170.64	173.32	**105.51**

[b] Perplexity is calculated based on the same valid set in Table 1

Correlation Analysis of AM Modeling BERT models trained on data of various sizes are compared against the best-performing SVD implementation in terms of both the Pearson and Spearman correlation w.r.t the human judgements on the system level. The results are presented in table. 3. The best model, BERT-20K, outperforms the best SVD implementation by 0.5 percent and 36.75 percent in terms of Pearson and Spearman correlation respectively. As the training data size increases to 50k and 100k, there is a drop in the performance. This may be due to the property of test twitter dialogues as well as the responses generated by the dialogue systems. Almost all the dialogues are between customer supports and users. In many test cases, the responses are very standard and lack semantic variation. For example, "*you are welcome.*" and "*thank you !*" are commonly-generated responses. Even human judges found it hard to rate such responses. In this case, more training data may not help improve the model's correlation w.r.t human judgements since the goal of a good adequacy metric implementation is to better represent the semantics of sentences. However, for the above-mentioned responses with little semantic variation, the AM model will find it hard to distinguish them even though there are more data to help improve its representational capability. Nonetheless, Deep AM-FM mitigates this problem by providing a more balanced judgement based on both adequacy and fluency and this is demonstrated in the later part of this section.

Table 3: AM-SVD vs AM-BERT in terms of system-level correlation w.r.t human judgements

Model	Pearson correlation	Spearman correlation	p-value
AM-SVD	0.8757	0.3970	4.23e-7*
BERT-10K	0.6815	0.1233	9.35e-4*
BERT-20K	**0.8802**	**0.5429**	3.09e-7*
BERT-50K	0.7905	0.1443	3.34e-5*
BERT-100K	0.7511	0.2511	1.35e-4*

p-value with asterisk indicates statistical significance (normally p-value should be < 0.05)

Correlation Analysis of FM Modeling The Pearson and Spearman correlation of various n-gram LMs (trained with full training data) and LSTM-RNN LMs (trained with different data sizes) are compared. The experimental results are presented in Table. 4. The best Pearson correlation is achieved by LSTM-RNN trained on 100K data at 0.9008 with a 2 percent improvement than the best baseline (4-gram LM). The best Spearman correlation is achieved by LSTM-RNN trained on 20K data at 0.6256 with a 31.6 percent improvement as compared to the best baseline (tri-gram LM). It is observed that the Pearson correlation progressively increases as the training data increases. This indicates that the deep learning implementation can address the limits of n-gram language models to provide more accurate judgement leveraging its ability to capture long-term contextual information with the aid of more data.

Table 4: N-gram vs LSTM-RNN in terms of system-level correlation w.r.t human judgements

Model	Pearson correlation	Spearman correlation	p-value
uni-gram	0.8128	0.1925	1.33e-5*
bi-gram	0.8596	0.2872	1.20e-7*
tri-gram	0.8272	0.4752	6.83e-6*
4-gram	0.8832	0.4331	2.50e-7*
5-gram	0.8820	0.3940	2.73e-7*
LSTM-RNN (10K)	0.6605	0.5880	1.52e-3*
LSTM-RNN (20K)	0.7408	**0.6256**	1.87e-4*
LSTM-RNN (50K)	0.7953	0.5985	2.77e-5*
LSTM-RNN (100K)	**0.9008**	0.5338	6.12e-8*

p-value with asterisk indicates statistical significance (normally p-value should be < 0.05)

Combining Deep AM-FM The correlation results of combining deep AM & FM components are compared against word-overlap metrics such as: BLEU, CiDER and ROUGE-L, and embedding-based metrics such as: skip-thought and embedding average as well as the original best AM-FM combination in Table 5. It can be observed that word-overlap metrics correlate poorly to human judgements in terms of both the Pearson and Spearman correlation. This is because word-overlap metrics are based on the assumption that there is significant overlap of words between

good responses and the corresponding golden references. However, conditioning on a given dialogue context, responses which are diverse in their usage of words and syntactic structures can be valid. The embedding-based metrics are better than their word-overlap counterparts in terms of correlation w.r.t human evaluation. However, they only evaluate along one dimension, the semantic closeness of the generated responses to the respective golden references, i.e. they do not account for the quality of the response construction.

Following [7], AM and FM scores are linearly combined by using the formula: $AM_{score} * \lambda + (1 - \lambda) * FM_{score}$ at system level, where λ is optimized on a development set to range between 0 and 1. λ reflects the relative emphasis on the adequacy and fluency dimension. Experimental results suggest that AM-FM framework exhibits high correlation w.r.t human evaluation, especially the deep learning based implementation, which achieves the best Pearson correlation of 0.9068 when $\lambda = 0.5$ and the best Spearman correlation of 0.5714 when $\lambda = 0.7$. In the original implementation, the best λ was empirically found to be 0.8, with a huge emphasis on the adequacy component. Deep AM-FM shifts the evaluation to a more balanced view with more or less equal emphasis on both the adequacy and fluency components. It is observed that as compared to the AM-FM baseline, an 1.8 percent improvement in terms of Pearson correlation is achieved when $\lambda = 0.5$ and there is a 29.2 percent gain when *lambda* is 0.7. This is especially helpful in the situation where one dimension of evaluation is insufficient to provide accurate judgement, then the other dimension will serve as an additional gauge to aid the distinguishing power of the model.

Table 5: Combined Deep AM-FM vs Other Metrics

Automatic Metric	Pearson Correlation	Spearman Correlation	p-value
BLEU-4	-0.5108	-0.1880	2.14e-2*
METEOR	0.3628	0.0692	1.16e-1
ROUGE-L	0.1450	0.0541	5.42e-1
CIDEr	-0.1827	0.2511	4.41e-1
Skip-Thoughts	-0.4608	-0.3549	4.09e-2*
Embedding Avg.	0.7747	0.0752	6.07e-5*
Vector Extrema	0.2250	0.0571	3.40e-1
Greedy Matching	0.3481	0.0060	1.33e-1
AM-FM Baseline	0.8907	0.4421	1.41e-7*
Deep AM-FM ($\lambda = 0.7$)	0.9005	**0.5714**	6.42e-8*
Deep AM-FM ($\lambda = 0.5$)	**0.9068**	0.5158	3.57e-8*

p-value with asterisk indicates statistical significance (normally p-value should be < 0.05)

Qualitative Analysis of Deep AM-FM Two sample dialogue contexts from the twitter test set are presented in Table 6. For each context, three hypotheses are listed: one with high human rating, one with low human rating and a generic/dull response. Scores provided by both human and deep AM-FM framework are also listed. It can be observed that deep AM-FM framework is able to distinguish the low-quality hy-

pothesis from the good ones as demonstrated by the positive correlation between human ratings and the deep AM-FM scores for the corresponding hypotheses. Individually, both the deep AM and deep FM component have the distinguishing capability. Interestingly, deep FM is able to provide more discriminative scores between the good and the bad hypotheses. This may be due to that the training of fluency component is based on the objective of minimizing perplexity. The model will be more confident in generating next tokens by minimizing the perplexity. Hence, a good language model is able to reasonably assess the confidence of different hypotheses conditioning on the given context and thus, better discriminates different hypotheses. Recently, it is reported in [25] that the objective of minimizing perplexity has strong correlation with their proposed sensibleness-specificity metric. This corroborates the idea of improving the language model in the fluency-metric module helps improve the effectiveness of the evaluation. It is also mentioned in [25] that measuring along the dimension of sensibleness alone tend to favor dull responses, such as *yes* and *I don't know*, which are safe answers, but not specific to the context. We provide two separate dull responses to the two sample dialogue contexts and their corresponding deep AM-FM scores to examine the framework's effectiveness in such situations. It can be observed that deep AM-FM gives low scores to the non-specific responses. This may be because deep AM-FM provides relative scores instead of absolute model-generated values. With the presence of gold references, which are specific to the context, comparisons between different hypotheses and respective references will help avoid favoring the dull responses.

Table 6: Dialogue Evaluation Samples

Context	Hypothesis	Avg. Human Rating	Deep AM Score	Deep FM Score	Deep AM-FM Score
U: gallery of images taken using @getnarrative during a recent visit to the mustang holding facility in burns , oregon .	great shot , <USER>! thank you for sharing this with us .	4.2	0.908	0.572	0.740
	hi, <USER>, we 're sorry to hear this . please dm us your contact details so we can look into this for you . thanks .	2.2	0.723	0.279	0.501
	Yes!	N.A	0.470	0.274	0.372
U: continues to point fingers and blame instead of fixing fridge after months. <URL>	we 're sorry for the frustrations . please dm your contact info along with mod / serial # . we can look further into this for you .	4.2	0.843	0.878	0.861
	we know tons about refrigerator repair. have a look! 20% off for a limited time: <URL>	2.5	0.839	0.796	0.818
	I don't know!	N.A	0.329	0.141	0.235

6 Conclusion & Future Work

In this paper, we propose deep AM-FM, a toolkit for automatic dialogue evaluation leveraging deep learning techniques. The purpose of the paper is to showcase the feasibility of applying different methodologies for modeling the adequacy metric and fluency metric so as to better adapt to the evaluation tasks. We demonstrate deep learning's ability to address the problems of the original latent semantic indexing and n-gram language model implementation and leverage the power of data to provide better evaluation. Currently, the toolkit is still at its initial version and we aim to consistently improve deep AM-FM and make it a common platform for evaluating text generation tasks in NLP, such as machine translation, dialogue system and text summarization. We will conduct more experiments and analyses with various deep learning techniques on more evaluation datasets. Most importantly, we aim to incorporate the pragmatics component (PM) into the original formulation of AM-FM framework to account for other aspects of the evaluation (to mimic human judgements). For example, in the dialogue setting, aspects like dialogue coherence, system's ability to provide consistent dialogue and ability to understand subtle cues in user's queries will be considered.

Acknowledgements This research is carried out under the collaboration program between Electrical & Computer Engineering Department, National University of Singapore and Robert Bosch (SEA) Pte Ltd. This research is also supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-GC-2019-002) as well as the Spanish projects AMIC (MINECO, TIN2017-85854-C4-4-R) and CAVIAR (MINECO, TEC2017-84593-C2-1-R).

References

1. Banchs, R. E., D'Haro, L. F., & Li, H. (2015). Adequacy–fluency metrics: Evaluating MT in the continuous space model framework. IEEE/ACM TASLP, 23(3), 472-482.
2. Bojanowski, P., Grave, E., Joulin, A., et al. (2017). Enriching word vectors with subword information. Transactions of ACL, 5, 135-146.
3. Chelba, C., Mikolov, T., Schuster, et al. (2014). One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. In Interspeech 2014.
4. Cho, K., Van Merriënboer, B., Gulcehre, C., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
5. Dai, Z., Yang, Z., Yang, Y., et al. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860.
6. Devlin, J., Chang, M. W., Lee, K., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL 2019: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).
7. D'Haro, L. F., Banchs, R. E., Hori, C., & Li, H. (2019). Automatic evaluation of end-to-end dialog systems with adequacy-fluency metrics. Computer Speech & Language, 55, 200-215.
8. Hill, F., Cho, K., Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. arXiv preprint arXiv:1602.03483.
9. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

10. Hori, C., & Hori, T. (2017). End-to-end conversation modeling track in DSTC6. arXiv preprint arXiv:1706.07440.
11. Kiros, R., Zhu, Y., Salakhutdinov, R. R., et al. (2015). Skip-thought vectors. In Advances in neural information processing systems (pp. 3294-3302).
12. Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284.
13. Liu, C. W., Lowe, R., Serban, I. V., et al. (2016). How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. EMNLP 2016 (pp. 2122-2132).
14. Kudo, T., Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. arXiv preprint arXiv:1808.06226.
15. Jozefowicz, R., Vinyals, O., Schuster, M., et al. (2016). Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410.
16. Logeswaran, L., Lee, H. (2018). An efficient framework for learning sentence representations. arXiv preprint arXiv:1803.02893.
17. Marelli, M., Bentivogli, L., Baroni, M., et al. (2014). Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. SemEval 2014 (pp. 1-8).
18. Mei, H., Bansal, M., Walter, M. R. (2017, February). Coherent dialogue with attention-based language models. In Thirty-First AAAI Conference on Artificial Intelligence.
19. Mikolov, T., Karafiát, M., Burget, et al. (2010). Recurrent neural network based language model. InterSpeech 2011.
20. Palangi, H., Deng, L., Shen, Y., et al. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM TASLP, 24(4), 694-707.
21. Peters, M. E., Neumann, M., Iyyer, M., et al. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
22. Radford, A., Wu, J., Child, R., et al. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8).
23. Sundermeyer, M., Schlüter, R., Ney, H. (2012). LSTM neural networks for language modeling. Interspeech 2012.
24. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
25. Adiwardana, D., Luong, M. T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., ... Le, Q. V. (2020). Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977.