Sequence to Backward and Forward Sequences: A Content-Introducing Approach to Generative Short-Text Conversation

Lili Mou,[1] **Yiping Song,**[2] **Rui Yan,**[3] **Ge Li,**[1,*] **Lu Zhang,**[1] **Zhi Jin**[1,*]
[1]Key Laboratory of High Confidence Software Technologies (Peking University), MoE, China
Institute of Software, Peking University, China   *Corresponding authors
[2]Institute of Network Computing and Information Systems, Peking Univerity, China
[3]Institute of Computer Science and Technology, Peking University, China
doublepower.mou@gmail.com
{songyiping,ruiyan,lige,zhanglucs,zhijin}@pku.edu.cn

Abstract

Using neural networks to generate replies in human-computer dialogue systems is attracting increasing attention over the past few years. However, the performance is not satisfactory: the neural network tends to generate safe, universally relevant replies which carry little meaning. In this paper, we propose a content-introducing approach to neural network-based generative dialogue systems. We first use pointwise mutual information (PMI) to predict a noun as a keyword, reflecting the main gist of the reply. We then propose `seq2BF`, a "*sequence to backward and forward sequences*" model, which generates a reply containing the given keyword. Experimental results show that our approach significantly outperforms traditional sequence-to-sequence models in terms of human evaluation and the entropy measure, and that the predicted keyword can appear at an appropriate position in the reply.

1 Introduction

Automatic human-computer conversation is a hot research topic in natural language processing (NLP). In past decades, researchers have developed various rule- or template-based systems, which are typically in vertical domains, e.g., transportation (Ferguson et al., 1996) and education (Graesser et al., 2005). In the open domain, data-driven approaches play an important role, because the diversity and uncertainty make it virtually impossible for humans to design rules or templates. Isbell et al. (2000) and Wang et al. (2013) use information retrieval methods to search for a reply from a pre-constructed database; Ritter et al. (2011) formalize conversation as a statistical machine translation task.

Recently, the renewed prosperity of neural networks brings new opportunities to open-domain conversation (Vinyals and Le, 2015; Shang et al., 2015; Serban et al., 2016a; Li et al., 2016a). In these studies, researchers leverage sequence-to-sequence (`seq2seq`) models to encode a *query* (user-issued utterance) as a vector and to decode the vector into a *reply*. In both encoders and decoders, an RNN keeps one or a few hidden layers; at each time step, it reads a word and changes its state accordingly. RNNs are believed to be well capable of modeling word sequences, benefiting machine translation (Sutskever et al., 2014), abstractive summarization (Rush et al., 2015) and other tasks of natural language generation. Contrary to retrieval methods, neural network-based conversation systems are *generative* in that they can synthesize new utterances; results in the literature also show the superiority of `seq2seq` to phrase-based machine translation for dialogue systems (Shang et al., 2015). In our study, we focus on neural network-based generative *short-text conversation*, where we do not consider context information, following Wang et al. (2013) and Shang et al. (2015).

Despite these, neural networks' performance is far from satisfactory in human-computer conversation. A notorious problem is the *universal reply*: the RNN prefers to generate safe, universally relevant sentences with little meaning, e.g., "something" (Serban et al., 2016a) and "I don't know" (Li et al., 2016a). One problem may lie in the objective of decoding. If we choose a reply with the maximal estimated

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3349–3358, Osaka, Japan, December 11-17 2016.

probability (either greedily or with beam search), it is probable to obtain such universal replies, because they do appear frequently in the training set. Another potential problem is that, the query may not convey sufficient information for the reply, and thus the encoder in `seq2seq` is less likely to obtain an informative enough vector for decoding.

In this paper, we propose a content-introducing approach to generative short-text conversation systems, where a reply is generated in a two-step fashion: (1) First, we predict a keyword, that is, a noun reflecting the main gist of the reply. This step does not capture complicated semantic and syntactic aspects of natural language, but estimates a keyword with the highest pointwise mutual information against query words. The keyword candidates are further restricted to nouns, which are not as probable as universal words (e.g., *I* and *you*), but can introduce substantial content to reply generation. (2) We then use a modified encoder-decoder model to synthesize a sentence containing the keyword. In traditional `seq2seq`, the decoder generates the reply from the first word to the last in sequence, which prevents introducing certain content (i.e., a given word) to the reply. To tackle this problem, we propose `seq2BF`, a novel "*sequence to backward and forward sequences*" model, based on our previous work of backward and forward language modeling (Mou et al., 2015). The `seq2BF` model decodes a reply starting from a given word, and generates the remaining previous and future words subsequently. In this way, the predicted keyword can appear at an arbitrary position in the generated reply.

The rest of this paper is organized as follows. Section 2 describes the proposed approach; Section 3 presents experimental results. Section 4 briefly reviews related work in the literature. Finally we conclude our paper and discuss future work in Section 5.

2 Our Approach

In this section, we present our content-introducing generative dialogue system in detail. Subsection 2.1 provides an overview of our approach, Subsection 2.2 introduces the keyword predictor, and Subsection 2.3 elaborates the proposed `seq2BF` model. We describe training methods in Subsection 2.4.

2.1 Overview

Figure 1 depicts the overall architecture of our approach, which comprises two main steps:

Step I: We first use PMI to predict a keyword for the reply, as shown in Figure 1a.

Step II: After keyword prediction, we generate a reply conditioned on the keyword as well as the query. More specifically, we propose the `seq2BF` model, which generates the backward half of the sequence (Figure 1b) and then the forward half (Figure 1c).

Notice that, the RNNs in Step II do not share parameters (indicated by different colors in the figure) because they differ significantly from each other. Moreover, the encoder and decoder do not share parameters either, which is standard in `seq2seq`. For clarity, we do not assign different colors for encoders and decoders, but separate them with a long arrow in Figures 1b and 1c.

2.2 Keyword Prediction

In this step, we use pointwise mutual information (PMI) to predict a keyword for the reply. We leverage such surface statistics because this step outputs a single keyword, which does not capture complicated syntax and semantics of queries and replies. Our goal of content introducing is to suggest a word that is especially suited to the query, instead of predicting a most likely (common) word. Hence, the pointwise mutual information is an appropriate statistic for keyword prediction.

Formally, we compute PMI of a query word w_q and a reply word w_r using a large training corpus by

$$\mathrm{PMI}(w_q, w_r) = \log \frac{p(w_q, w_r)}{p(w_q)p(w_r)} = \log \frac{p(w_q|w_r)}{p(w_q)} \tag{1}$$

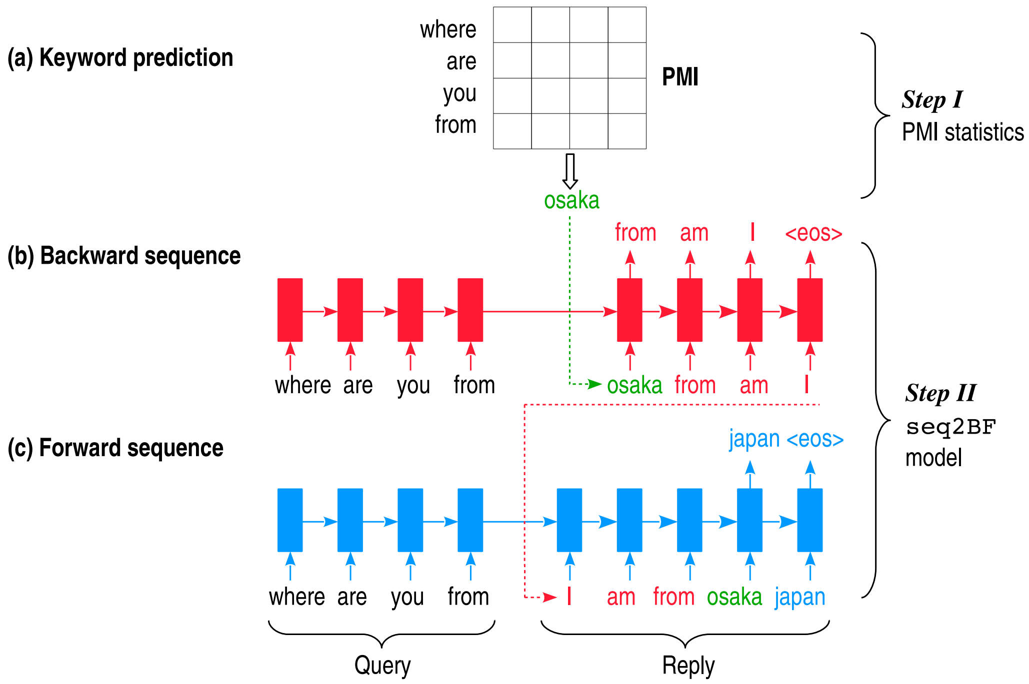

Figure 1: An overview of our content-introducing approach to generative dialogue systems.

When predicting, we choose the keyword w_r^* with the highest PMI score against query words $w_{q_1}, \cdots w_{q_n}$, i.e., $w_r^* = \operatorname{argmax}_{w_r} \text{PMI}(w_{q_1} \cdots w_{q_n}, w_r)$, where

$$\text{PMI}(w_{q_1} \cdots w_{q_n}, w_r) = \log \frac{p(w_{q_1} \cdots w_{q_n} | w_r)}{p(w_{q_1} \cdots w_{q_n})} \tag{2}$$

$$\approx \log \frac{\prod_{i=1}^n p(w_{q_i} | w_r)}{\prod_{i=1}^n p(w_{q_i})} = \sum_{i=1}^n \log \frac{p(w_{q_i} | w_r)}{p(w_{q_i})} = \sum_{i=1}^n \text{PMI}(w_{q_i}, w_r) \tag{3}$$

The approximation is due to the independency assumptions of both the prior distribution $p(w_{q_i})$ and posterior distribution $p(w_{q_i} | w_r)$. While the two assumptions may not be true, we use them in a pragmatic way so that the word-level PMI is additive for a whole utterance. Experiments show that this treatment generally works well.

Different from choosing the most likely word, PMI penalizes a common word by dividing its prior probability; hence, PMI prefers a word that is most "mutually informative" with the query. Moreover, we manually restrict keyword candidates to nouns, so that this step can introduce substantial content to reply generation, which will be discussed in the next part.

2.3 The `seq2BF` Model

To insert the predicted keyword into sequence generation, we cannot use the traditional `seq2seq` model. In existing approaches, we usually decompose the probability of an output sentence $\boldsymbol{r} = r_1 r_2 \cdots r_m$ given an input sentence $\boldsymbol{q} = q_1 q_2 \cdots q_n$ by

$$p(r_1, \cdots, r_m | \boldsymbol{q}) = p(r_1 | \boldsymbol{q}) p(r_2 | r_1, \boldsymbol{q}) \cdots p(r_m | r_1 \cdots r_{m-1}, \boldsymbol{q}) = \prod_{i=1}^m p(r_i | r_1 \cdots r_{i-1}, \boldsymbol{q}) \tag{4}$$

The output sentence is thus predicted in sequence from r_1 up to r_m either greedily or with beam search. I personally believe such decomposition is mainly inspired by the observation that humans always say a sentence from the beginning to the end.

However, in our content-introducing approach to generative dialogue systems, the predicted keyword could appear at the beginning (r_1), the middle (r_2 to r_{m-1}), or the end (r_m) of the reply. It is then natural to decompose the probability starting from the given word. In particular, the predicted keyword k splits a reply into two (sub-)sequences:

$$\begin{aligned}\text{Backward sequence:} \quad & r_{r-1}, \cdots, r_1 \\ \text{Forward sequence:} \quad & r_{k+1}, \cdots, r_m\end{aligned}$$

and the joint probability of remaining words can be written as

$$p\left(\frac{r_{k-1}\cdots r_1}{r_{k+1}\cdots r_m}\middle| r_k, \boldsymbol{q}\right) = \prod_{i=1}^{k-1} p^{(\text{bw})}(r_{k-i}|r_k, \boldsymbol{q}, \cdot) \prod_{i=1}^{m-k} p^{(\text{fw})}(r_{k+i}|r_k, \boldsymbol{q}, \cdot) \tag{5}$$

where $p(\text{- - -}|r_k, \boldsymbol{q})$ refers to the probability of the backward and forward subsequences given the split word r_k and an encoded query $\boldsymbol{q}$. Notice that both the backward and forward sequence generators include a wildcard allowing rich inner-subsequence and/or inter-subsequence dependencies. In our previous study of backward-and-forward (B/F) language modeling (Mou et al., 2015), we propose three variants: (1) **sep-B/F**: The backward and forward sequences are generated separately. (2) **syn-B/F**: The backward and forward sequences are generated synchronously using a single RNN, two output layers at each time step for the two sequences. (3) **asyn-B/F**: The two sequences are generated asynchronously, that is, we first generate the backward "half" sequence, conditioned on which we then generate the forward "half." Our previous experiments show the asyn-B/F is the most natural way of modeling backward and forward sequences, and thus we adopt this variant in `seq2BF`.

Specifically, our `seq2BF` model works as follows. A `seq2seq` neural network encodes a query and decodes a "half" reply, that is, the first set of factors in Equation 5 becomes $p^{(\text{bw})}(r_{k-i}|r_k, \boldsymbol{q}, \cdot) = p^{(\text{bw})}(r_{k-i}|r_k \cdots r_{k-i+1}, \boldsymbol{q})$, where $1 \le i \le k-1$. The decoder here outputs words in a reversed order from r_{k-1}, r_{k-2} to r_1, so that the reversed "half" sequence is fluent with respect to the given word, at least from a mathematical perspective. (Please see Figure 1b.)

Then another `seq2seq` model encodes the query again, but decodes the entire reply, provided that the first half of the reply is given (Figure 1c), i.e., $p^{(\text{fw})}(r_{k+i}|r_k, \boldsymbol{q}, \cdot) = p^{(\text{fw})}(r_{k+i}|r_1 \cdots r_k \cdots r_{k+i-1}, \boldsymbol{q})$, $(1 \le i \le m-k)$. Here, the forward generator is aware of the backward half sequence $r_1 \cdots r_{k-1}$, where the word order is reversed again, so that they are in a normal order for fluent forward generation.

In both backward and forward `seq2seq` models, we use RNNs with gated recurrent units (GRUs) for information processing (Cho et al., 2014), given by

$$\boldsymbol{r}_t = \sigma(W_r \boldsymbol{w}_t + U_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r) \tag{6}$$

$$\boldsymbol{z}_t = \sigma(W_z \boldsymbol{w}_t + U_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_z) \tag{7}$$

$$\widetilde{\boldsymbol{h}}_t = \tanh\Big(W_h \boldsymbol{w}_t + U_h(\boldsymbol{r} \circ \boldsymbol{h}_{t-1}) + \boldsymbol{b}_h\Big) \tag{8}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \circ \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \circ \widetilde{\boldsymbol{h}}_t \tag{9}$$

where W's and U's are weights and $\boldsymbol{b}$'s are bias terms. $\boldsymbol{w}_t$ is the word embedding; $\boldsymbol{h}_t$ is the hidden state at the time step t. "$\circ$" denotes element-wise product.

2.4 Model Training

Training (i.e., parameter estimation) is always a most important thing in the neural network regime, and oftentimes, problems arise when we prepare the dataset.

Fortunately, the `seq2BF` model can be trained without additional labels. We randomly sample a word in a reply as the split word, take the first half, and reverse its word order; in this way, we obtain the training data for the backward sequence generator. The forward sequence generator is essentially a `seq2seq` encoder and decoder from queries to replies. The difference between the pure `seq2seq` and the forward generator of `seq2BF` lies in the inference stage: in our scenario, we ignore the query-reply `seq2seq` generator's output at the beginning steps, but feed it with the "half" reply obtained by our backward sequence generator as well as the predicted keyword (red and green words in Figure 1c); then we let the `seq2seq` model generate remaining future words (blue words in Figure 1c).

It should be emphasized that the backward sequence generator requires "half" replies starting from the split word as training data, and that we cannot train the model with a full reversed sentence. Otherwise, the backward part will undesirably generate an entire reversed reply, and the forward part cannot add much to it.

3 Experiments

3.1 Dataset

We evaluated our approach on a Chinese dataset of human conversation crawled from the Baidu Tieba[1] forum. We used 500,000 query-reply pairs to train the `seq2BF` model. We had another unseen 2000 and 27,871 samples for validation and testing, respectively. To obtain PMI statistics in the first step (Figure 1a) of our method, we use a much larger dataset containing 100M query-reply pairs.

Chinese language is different from English in that a Chinese character carries more semantics than an alphabet. For example, the characters 黑 and 板 mean "black" and "board" in English, respectively; the term 黑板 means "blackboard." Because we have far more Chinese terms than English words, our `seq2BF` is trained in the character level out of efficiency concerns. But we train the keyword predictor with noun phrases (Chinese terms), by noticing that *blackboard* is different from *board*, despite some subtle relations. Fortunately, the two granularities can be integrated together straightforwardly: during backward sequence generation, we only need to condition the model on the character sequence in the key term instead of a single keyword, that is, we have several green inputs in Figure 1b. In our study, we kept 2.5k noun terms as candidate keywords and 4k characters for `seq2BF` generation.

3.2 Hyperparameters

In our experiments, word embeddings and recurrent layers were 500d. We used rmsprop to optimize all parameters except embeddings, with initial weights uniformly sampled from $[-.08, .08]$, initial learning rate 0.002, moving average decay 0.99, and a damping term $\epsilon = 10^{-8}$. Because word embeddings are sparse in use (Peng et al., 2015), we optimized embeddings asynchronously by stochastic gradient descent with the learning rate divided by $\sqrt{\epsilon}$. We set the mini-batch size to 50. These values were mostly chosen empirically by following Karpathy et al. (2015) and Mou et al. (2015); they generally work well in our scenarios. We did not tune the hyperparameters in this paper, but are willing to explore their roles in dialogue generation as future work.

The validation set (containing 2k query-reply samples) was used for early stop only. We chose the parameters yielding the highest character-level BLEU-2 score on our validation set.

3.3 Performance

We evaluated our results in terms of the following criteria:

- **Human Evaluation**. We had six volunteers[2] to annotate the results of our content-introducing `seq2BF` and baselines. To ensure the quality of human evaluation, we randomly sampled 200 queries and replies in the test set. The samples and volunteers were further split into two equal-sized groups of different annotation protocols:
  - *Pointwise annotation*. The volunteers were asked to annotate a score indicating the appropriateness of a reply to a given query: 0 = bad reply, 2 = good reply, and 1 = borderline.
  - *Pairwise annotation*. Given a certain query, the volunteers were asked to judge whether pure `seq2seq` is better than, equal to, or worse than `seq2BF` (with content introducing). If they could not understand both replies, they were asked to choose "equal."

  All our human evaluation were conducted in a random, blind fashion, i.e., we randomly shuffled the samples and volunteers did not know which system generated a particular reply. Notice that we did not define what a "good" or "bad" reply is; otherwise, the annotation may be biased towards certain systems. Rather, annotators had their own subjective discretion. While criteria may differ from

[1] http://tieba.baidu.com
[2] All volunteers are well-educated native speakers of Chinese and have received a Bachelor's degree or above.

(a)

Method	PointHuman	Length	Entropy
seq2seq	0.58	**5.61**	6.960
seq2BF$_-$	0.46	5.60	6.971
seq2BF$_+$	**0.67**	5.31	**9.139**
Groundtruth	–	9.19	8.832

(b)

	PairHuman		
Method	Wins	Ties	Loses
seq2seq	24.7	26.0	49.3
seq2BF$_+$	**49.3**	26.0	24.7

Table 1: The performance of our content-introducing seq2BF (denoted as seq2BF$_+$) dialogue system in comparison with pure seq2seq and seq2BF without predicted keywords (seq2BF$_-$). **PointHuman**: Pointwise human evaluation. (Annotator agreement: std=0.33, Fleiss' κ=0.27.) **PairHuman**: Pairwise human evaluation, which shows the percentage at which a system wins, loses, or ties in comparison with the other (κ=0.29).

person to person, the ranking of average scores reflects the comparison of different dialog systems. (In our study, the ranking is consistent among all six annotators.)

- **Length**. The length of an utterance is an objective, surface metric that reflects the substance of a generated reply.
- **Entropy**. Entropy is another objective metric, which shows the serendipity of a reply by measuring the amount of information contained in the utterance. We computed the average character-level entropy, given by

$$-\frac{1}{|R|}\sum_{w\in R}\log_2 p(w) \tag{10}$$

where R refers to all replies, $|R|$ is the number of words in all replies, and $p(\cdot)$ is the unigram probability of a character in the training set.

The latter two metrics are "intrinsic," by which we mean no reference (groundtruth reply) is needed to compute the metric. They are used in Serban et al. (2016b).[3] In our experiments, objective metrics were assessed with all test samples.

We do not include BLEU scores as our evaluation criteria, which are used in Li et al. (2016a). As a pilot study, we also asked two volunteers to write their own replies to 50+ queries. One obtained 1.69 BLEU-4 score. (The result is lower than 1.74 obtained by an automatic dialogue system in Li et al. (2016a); this may be caused by different datasets and languages.) What surprises us is that the other volunteer obtained 0 BLEU-2 score, indicating that no bi-gram overlaps between his replies and the references. This result provides evidence of the diversity among human replies, and thus we abandoned BLEU scores as evaluation criteria. We reported this case study in our paper so as to shed more light on the research of evaluation metrics in dialogue systems.

Table 1 compares our content-introducing seq2BF with the seq2seq baseline. Also included is the seq2BF model without content introducing, that is, we slacked the keyword constraint in the backward sequence generator (Figure 1b), and let it generate a reversed half reply freely.

As seen in Table 1a, the content-introducing seq2BF model yields the highest pointwise human annotation result. For seq2BF without content introducing, we obtain low scores, showing that artificially splitting a sequence into two parts itself is not a fancy way of modeling natural language sentences. However, given a keyword predicted by PMI statistics, the backward and forward sequence generation can significantly improve the dialogue system in comparison with pure seq2seq. In the pairwise annotation protocol, the content-introducing seq2BF model also outperforms traditional seq2seq to a large extent. We computed standard deviation (0.33) and discrete-class Fleiss' κ scores (0.27–0.29), showing fair annotator agreement (Fleiss, 1971). While the agreement is comparable to previous results, e.g., 0.2–0.4 reported in Shang et al. (2015), we find the disagreement mainly comes from the discretion of

[3]The entropy equation in Serban et al. (2016b, v3) has a minor error. We confirmed this with Serban via personal emails.

	Chinese	English (translated)
Query	李有男友公开过了	It's known that *Li*† has a boyfriend.
Grountruth	都已经分了 之前李的贴吧都在讨论了	Broken up. There's discussion in her *Tieba*.‡
seq2seq	我是男的	I am a male
seq2BF$_-$	我的男友	My boyfriend
seq2BF$_+$	有**绯闻**男友	Has a **rumor**ed boyfriend
Query	人大复试飘过	Passed second-round exam of Renming Univ.
Grountruth	这么牛，什么专业	Cool, what's your major
seq2seq	我也是！	Me too!
seq2BF$_-$	我也是！	Me too!
seq2BF$_+$	**分数**是什么	What is your **score**
Query	挺漂亮的 祝福祝福	So beautiful, congratulations
Grountruth	下手慢了哈。现在	You're late till now
seq2seq	谢谢	Thank you
seq2BF$_-$	是的，谢谢	Yes, thank you
seq2BF$_+$	谢谢**夸奖**	Thank you for **praising**
Query	我要换头像了！	Want to change a photo
Grountruth	打算换成什么啊～	What do you like...
seq2seq	我是你的头像	I'm in your photo
seq2BF$_-$	我是你的头像	I'm in your photo
seq2BF$_+$	第一张**图像**是谁	Who is in your first **photo**

Table 2: Examples of generated replies. Predicted keywords are in **bold**. †*Li* is a Chinese actress; we anonymized her first name. ‡*Tieba* is a Chinese forum where our datasets are crawled.

Model	seq2seq	seq2BF$_-$	seq2BF$_+$ keyword	seq2BF$_+$ remaining
Entropy	6.960	6.971	11.630	7.422

Table 3: Fine-grained analysis of character-level entropy. In the seq2BF$_+$ model, we analyze the average entropy of keywords and remaining words separately.

the "goodness" of a reply (so that the annotation complies with the subjective nature of human evaluation). All annotators yielded the same system-level ranking order, providing consistent evidence of the effectiveness of our approach.

Regarding intrinsic metrics, our seq2BF with a predicted keyword generates slightly shorter replies than seq2seq, but contains far richer information, as the entropy increases by 30%. The results verify that content introducing is particularly useful in generative human-computer dialogue systems.

3.4 Case Studies and Discussion

We provide case studies in Table 2.[4] As we see, the seq2seq responder prefers safe, universally relevant utterances like "me too." In these examples, the replies generally match the queries in meaning, but such universal replies are too boring and thus undesirable in real applications. In the seq2BF model with content introducing, we predict a keyword of the reply with PMI. This yields meaningful words/terms like *rumor* and *score*, serving as the gist of the reply. Then the seq2BF model generates previous and future words to obtain a more complete utterance (maybe not a whole sentence because of the casualness in human conversation). The proposed "backward and forward sequences" mechanism ensures that the predicted keyword can appear at an arbitrary position in the utterance.

We delve deep into the question: why seq2seq models (or variants) tend to generate universal

[4]Due to Chinese-English translation, some characteristics cannot be fully presented in the English text, e.g., the position of the given word, the length of the reply, and even the part-of-speech of a word. Nevertheless, we present the predicted keyword and its translated counterpart in bold and thus the aforementioned characteristics can be visually demonstrated to some extent.

replies? We may have two conjectures: (1) The `seq2seq` model cannot capture rich enough semantics other than "yes," "me too," etc. (2) The sequence generator is able to capture rich semantics, but starting from a high-level universal word at the beginning, it is unlikely to fall into concrete concepts.

We present in Table 3 the average character-level entropy of keywords and non-keywords. We find that, provided with a noun term, `seq2BF` can generate meaningful remaining words (keyword excluded) with an entropy of 7.422, higher than 6.971 given by `seq2BF` without keywords. Noticing that the `seq2BF` model is exactly the same in content-introducing and non-content-introducing settings, we believe the second conjecture holds. Choosing the most likely reply yields universal utterances; moreover, RNN sequence generators are reluctant to introduce concrete concepts, provided with one or a few universal words/terms (like *I* and *you*) that are greedily chosen at the beginning.

Fortunately, our content-introducing `seq2BF` works in an opposite fashion. We first predict a meaningful but not that probable noun term as the keyword; then we feed `seq2BF` with such concrete keyword that provides substantial content. In this way, our approach significantly outperforms pure `seq2seq` generation in short-text conversation systems.

4 Related Work

4.1 Dialogue Systems

Automatic human-computer conversation has long attracted the attention of researchers. In early decades, people design rule- or template-based systems, but they are mainly in vertical domains (Ferguson et al., 1996; Misu and Kawahara, 2007). Although such approaches can also be extended to the open domain (Han et al., 2015), their generated sentences are subject to 7 predefined forms and thus are highly restricted. For open dialogues, researchers have applied data-driven approaches, including retrieval methods (Isbell et al., 2000; Wang et al., 2013), phrase-based machine translation (Ritter et al., 2011), and recurrent neural networks (Sordoni et al., 2015; Shang et al., 2015).

A hot research topic in human-computer conversation is mixed-initiative systems, for example, the TRAINS-95 system for route planning (Ferguson et al., 1996) and AutoTutor for learner advising (Graesser et al., 2005). Li et al. (2016b) propose a proactive dialogue system that can introduce new content when a stalemate occurs. The system is chatbot-like and in the open domain; an external knowledge base is used for searching related entities as new content. They propose a random walk-like reranking algorithm based on retrieval results. Different from Li et al. (2016b)'s work, our paper addresses the problem of content introducing in open-domain generative dialogue systems.

4.2 Neural Networks for Sentence Generation

Sutskever et al. (2014) propose `seq2seq` for machine translation; the idea is to encode a source sentence as a vector by a recurrent neural network (RNN) and to decode the vector to a target sentence by another RNN. Bahdanau et al. (2015) enhance it with an attention mechanism. These approaches largely benefit natural language generation tasks such as abstractive summarization (Rush et al., 2015), question answering (Yin et al., 2016), and poetry generation (Wang et al., 2016).

For neural network-based dialogue systems, Sordoni et al. (2015) summarize a query and context as bag-of-words features, based on which an RNN decodes the reply. Shang et al. (2015) generate replies for short-text conversation by `seq2seq`-like neural networks with local and global attention. Yao et al. (2015) and Serban et al. (2016a) design hierarchical neural networks for multi-turn conversation.

To address the problem of universal replies, Li et al. (2016a) propose a mutual information training objective. Serban et al. (2016b) apply a variational Bayes approach that imposes a probabilistic distribution on the hidden variables and encodes the parameters of the posterior distribution. A very recent study similar to ours is Xing et al. (2016), where replies are augmented with topic information. In a dialogue-like question-answering system, Yin et al. (2016) query a knowledge base and insert a selected triple into an answer sentence by a soft logistic unit. In such approach, however, the answer may not actually appear in the generated sentence, especially when the test patterns are different from training ones. Unlike existing work, our `seq2BF` model guarantees the predicted keyword can appear in the reply at an appropriate position.

5 Conclusion and Future Work

In this paper, we proposed a content-introducing approach to generative short-text conversation systems. Instead of generating a reply sequentially from the beginning word to the end as in existing approaches, we used pointwise mutual information to predict a keyword, i.e., a noun term, for the reply. Then we proposed a "sequence to backward and forward sequences" (`seq2BF`) model to generate a reply containing the predicted keyword. The `seq2BF` mechanism ensures the keyword can appear at an arbitrary position in the reply, but the generated utterance is still fluent. Experimental results show that our approach consistently outperforms the pure `seq2seq` model in dialogue systems in terms of human evaluation and the entropy measure.

In future work, we would like to apply different keyword prediction techniques (e.g., neural sentence models) to improve the performance; the proposed `seq2BF` model can also be extended to other applications like generative question answering, where the answer may be given by searching an external database or knowledge base.

Acknowledgments

We thank Jiexiu Zhai, Hao Meng, Siqi Wang, Zhexi Hong, Junling Chen, and Zhiliang Tian for evaluating our dialogue systems. We also thank all reviewers for their constructive comments. This research is supported by the National Basic Research Program of China (the 973 Program) under Grant No. 2015CB352201 and the National Natural Science Foundation of China under Grant Nos. 61232015, 91318301, 61421091, 61225007, and 61502014.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of 8th Workshop on Syntax, Semtnatics and Structure in Statistical Translation*, pages 103–111.

George Ferguson, James Allen, and Brad Miller. 1996. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proceedings of Artificial Intelligence Planning Systems Conference*, pages 70–77.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378.

Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. 2005. AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.

Sangdo Han, Jeesoo Bang, Seonghan Ryu, and Gary Geunbae Lee. 2015. Exploiting knowledge base to generate responses for natural language dialog listening agents. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 129–133.

Charles Lee Isbell, Michael Kearns, Dave Kormann, Satinder Singh, and Peter Stone. 2000. Cobot in LambdaMOO: A social statistics agent. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 36–41.

Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Xiang Li, Lili Mou, Rui Yan, and Ming Zhang. 2016b. StalemateBreaker: A proactive content-introducing approach to automatic human-computer conversation. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2845–2851.

Teruhisa Misu and Tatsuya Kawahara. 2007. Speech-based interactive information guidance system using question-answering technique. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume IV, pages 145–148.

Lili Mou, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2015. Backward and forward language modeling for constrained natural language generation. *arXiv preprint arXiv:1512.06612*.

Hao Peng, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2015. A comparative study on regularization strategies for embedding-based neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2106–2111.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 583–593.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016a. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 3776–3783.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016b. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1577–1586.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 935–945.

Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese poetry generation with planning based neural network. In *Proceedings the 26th International Conference on Computational Linguistics*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic augmented neural response generation with a joint attention mechanism. *arXiv preprint arXiv:1606.08340*.

Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. 2015. Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565*.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2972–2978.