# Unsupervised Natural Language Generation for Task-Oriented Dialogue System

Dian Huang
Stanford University
dhaung05@stanford.edu

## Abstract

*Conventional natural language generation for task-oriented dialogue system requires labeled data, which can be expensive and error-prone. This paper proposed a dual-models approach for natural language generation of the task-oriented dialogue system, which use only label-free text data to generate grammatically correct sentence given keywords. The paper also demonstrates that the general language model can be applied to train a grammar model consisted of parts of speech without compromise on perplexity. The high ROUGE-l score on recall this model achieves on E2E dataset shows the success of the algorithm. The paper also presents a modified beam search for searching sentences with the keywords.*

## 1. Introduction

### 1.1. Related Work

Traditional task-oriented e-commerce dialogue system generates responses with fixed templates. It only needs to take the keywords and fill in the blank, but the response lacks variety on sentence structure [8]. Many techniques related to machine learning have been proposed to address these issues. For example, recurrent network model proposed in [9] uses encoder-decoder frame network to study the slot-to-value pair. A sequence to sequence model that can generate sentences from the keywords is also proposed in [2]. In spite of their promising results, they need the data that have the labels of keywords on the sentences, which can be error-prone and labor-intensive, as shown in table 1.

Various techniques have been published to improve natural language generation without labeled data. For example, K.Kumagai proposed the natural language generation with Monte Carlo Tree Search (MCTS) in [3]. However, the sentences it generates are short and have grammar errors. The generation also shows no understanding of its context. In [4], J. Li proposed using deep reinforcement in generating dialogue; however, the sentences generated are typically short and conversational. It may not be able to include all the key information given by the keywords in its generated sentences.

Therefore, this paper proposed a dual-models approach that no longer require any labeled data to generate a target sentence, but given the keywords, it is able to generate a grammatically correct sentence with all the keywords. As shown in table 1, the proposed method trains with only text data. The paper is organized as follows. Section 2 briefly describes the proposed approach. Section 3 presents the data sample and pre-processing. Section 4 describes the models. Section 5 presents the beam search. Section 6 and section 7 shows the evaluation and comparison. Section 8 and section 9 are the conclusions and future work.

Table 1. Comparison of data requirement in various approach

| Method | Labels | Text |
|---|---|---|
| Template | name[Zizzi], eatType[coffee shop], rating[average], location[Burger King] | The [Name] [eatType] is located near [location]. It has an [rating] customer rating. |
| Conventional Machine Learning | name[Zizzi], eatType[coffee shop], rating[average], location[Burger King] | The *Zizzi coffee shop* is located near *Burger King*. It has an *average* customer rating. |
| Proposed Approach | | The Zizzi coffee shop is located near Burger King. It has an average customer rating |

## 2. Algorithm Overview

As shown in figure 1, the proposed algorithm trains a grammar model to generate grammar rules consisted of parts of speech for the possible target sentences. The language model trained with only text data will output the

probability of each word. The beam search will then select the words that have the right parts of speech. Finally, the sentences with all the keywords and low perplexity will be selected as the response.

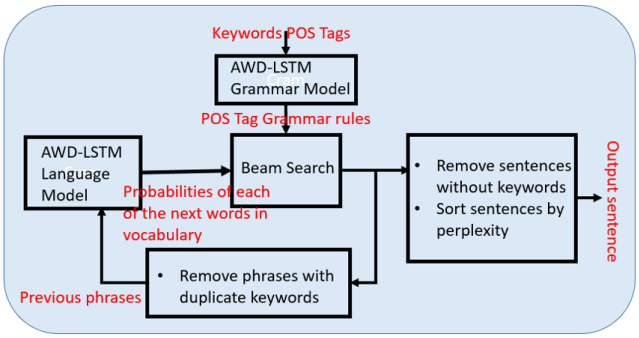


Figure 1. General architecture of proposed algorithm

## 3. Dataset

### 3.1. Data Collection

We collect the data from the E2E dataset [7] that has 51,426 instances of texts in the restaurant domain. Only the text portion of the data is used when training the model. The keywords in the E2E dataset are only used when testing the model. We randomly select 4000 samples to test our model. Some sample data are shown below:

Training Sample:
*The Dumpling Tree is an Italian restaurant with high prices.*
*The Phoenix is located in the city centre. It offers English food, has a high price range, and a customer rating of one out of five.*

Testing Sample:
*Keywords Input: Blue Spice, coffee shop, riverside*
*Reference: At the riverside, there is a coffee shop called The Blue Spice.*

### 3.2. Data Pre-processing

All number in the texts are replaced with the same random number. Stanford Part-Of-Speech Tagger (POS Tagger) [10] then reads the text and creates a grammar rule consisted of only the parts of speech. This grammar rule defines the sentence structure and refines the search of words. This creates a new dataset with POS tags only, as shown in Table 2. This tagger classifies the words into 39 categories, which will be used to train the grammar model.

Table 2. Sample POS Tag

| The | Zizzi | Coffee | shop | is | located | near | Burger | King |
|---|---|---|---|---|---|---|---|---|
| DT | NNP | NN | NN | VBZ | VBN | IN | NNP | NNP |
| There | is | a | pub | called | Zizi | near | the | Sorrento |
| EX | VBZ | DT | NN | VBN | NNP | IN | DT | NNP |

## 4. Models

### 4.1. AWD-LSTM

ASGD Weight-Dropped LSTM (AWD-LSTM) [5] has achieves the state-of-the-art word level perplexities of 52.8 on Penn Treebank and 52.0 on WikiText-2. ASGD stands for average stochastic gradient descent. LSTM stands for long short-term memory. Perplexity is the exponent of the cross-entropy loss, as shown in the equation below:

$$perplexity = 2^{\frac{1}{N}\sum_i -logP(s_i)} \tag{1}$$

where $s_i$ is the $i_{th}$ sentence, $P(s_i)$ is the probability of the sentence $s_i$, N is the number of sentences. We can express the probability of a sentence as:

$$P(s) = P(w_1|w_0)P(w_2|w_0, w_1)\dots P(w_n|w_1, \dots, w_{n-1}) \tag{2}$$

where $w_i$ is the $i_{th}$ words in the sentence. n is the number of words in a sentence. LSTM has demonstrated its superior capability in handling long-term dependency task. It is a special type of recurrent neural network. In this work, we use this network to train both grammar and language model. The mathematical formula of LSTM is shown as follows.

$$i_t = \sigma(W^i x_i + U^i h_{t-1}) \tag{3}$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1}) \tag{4}$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1}) \tag{5}$$

$$\widetilde{c}_t = tanh(W^c x_t + U^c h_{t-1} \tag{6}$$

$$c_t = i_t \odot \widetilde{c}_t + f_t \odot \widetilde{c}_{t-i} \tag{7}$$

$$h_t = o_t \odot tanh(c_t) \tag{8}$$

where $[W^i, W^f, W^o, U_i, U_f, U_o]$ are weight matrices, $x_t$ is the vector input to the timestep $t$, $h_t$ is the current exposed hidden state, $c_t$ is the memory cell state, and $\odot$ is element-wise multiplication. The memory cell helps to remember the information for a longer period. The gates control when to forget this information.

As shown in figure 2, AWD-LSTM applies Dropconnect on the hidden-to-hidden weight matrices to prevent overfitting, which drops a subset of weight of each node based on the probability, as a naïve drop-out can disrupt the long-term dependency of recurrent neural network (RNN), which is particularly important for this application as it needs to generate long-length sentence.

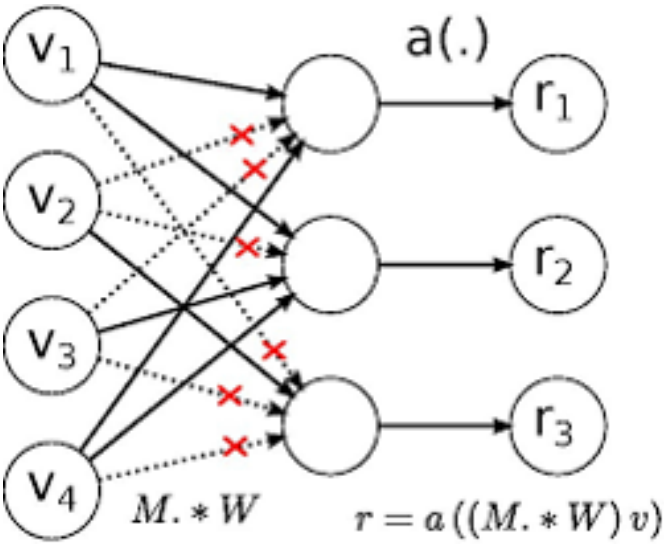


Figure 2. Dropconnect cuts the connection

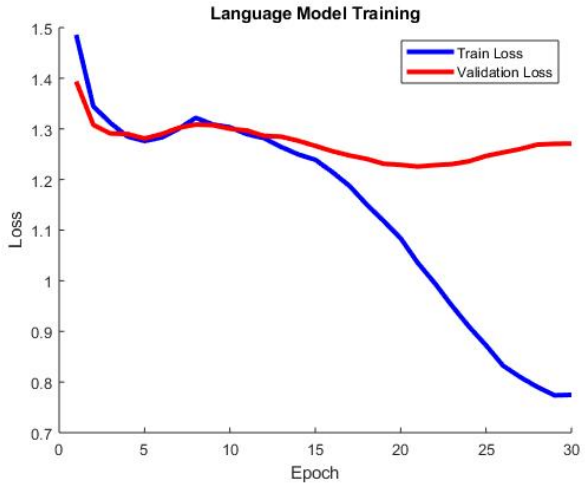


Figure 3. Training and validation cross entropy loss vs epoch

### 4.2. Language Model

For task-oriented dialogue system, it is necessary to narrow the scope in order to reduce the perplexity even further and to predict the next words related to the restaurant rather than a general field. Therefore, with transfer learning, the AWD-LSTM language on WikiText-2 is fine-tuned with the dataset about restaurant domain only. This fine-tuned model achieves a perplexity of only 3.56 on the E2E dataset. Figure 3 shows the training and validation loss. With small dataset about the restaurant, it is able to reach convergence much faster as the size of vocabulary and variation in sentence structure is much smaller. Given the first word as ”xxbo” (beginning of the sentence) and select the next n words with the highest probability, it is able to generate grammatically correct, fluent sentences, as shown below:

Predicted words after ”xxbos”:

*xxbos the pub The Waterman , serves Japanese food , is in the range of 229 - 229 and is highly rated by customers in the Riverside area xxbos Zizzi is a kid friendly coffee shop near the riverside .*

However, to search for a sentence with high probability that has all keywords in these generated sentences remains to be infeasible. If the keywords happen to be those words rarely occurred in the dataset, the sentence that has these keywords will have low probability. For example, ”the Chipotle” may have a lower probability than ”the Starbuck” depending on the dataset. Moreover, it has to search for the next word in the entire vocabulary. For example, ”the restaurant” may have a higher probability than those of ”the Chipotle” and ”the Starbuck”, although the latter is generally one of the keywords. So it will always search for the word ”restaurant” instead, as this word increases the probability of a sentence.

### 4.3. Grammar Model

To resolve this problem, this paper proposed an AWD-LSTM based grammar model that takes the part-of-speech tags as the training set. It generates a grammar rule for a sentence with part-of-speeches, as shown in table 4.3. There are two advantages to this grammar model. First, ”the Chipotle” and ”the Starbuck” have the same part of speech ”DT NNP”, where ”DT” is the determiner and ”NNP” is the proper noun. So their difference in occurrence does not affect the generation of the grammar rule. Second, it only needs to search from 39 different parts of speech, so the grammar model achieves a perplexity of 2.16. So applying beam search, later on, can find the sentence grammar rule that has all the POS of the keywords, as shown in 4.3:

Table 3. Generate POS of potential target sentences that have keywords POS: Must have keywords POS:”NNP”, ”NN”, NN”

| Key words | Zizi | Coffee | Shop | | | |
|---|---|---|---|---|---|---|
| POS | NNP | NN | NN | | | |
| DT | *NNP* | *NN* | *NN* | VBZ | DT | JJS |
| *NNP* | *NN* | *NN* | MD | VB | DT | JJS |

Predicted part-of-speech after ”xxbos”:

*xxbos DT NNP NNP , DT NN JJ NN , VBZ DT JJ NN IN DT JJ NN NN CC VBZ RB NN HYPH JJ .*

As shown in figure 4, the training loss of the grammar model is surprisingly higher than validation loss. One possible reason can be related to the dropout in the model. The other reason is that there may not be sufficient variety of the sentence structure in the dataset, so the training samples of

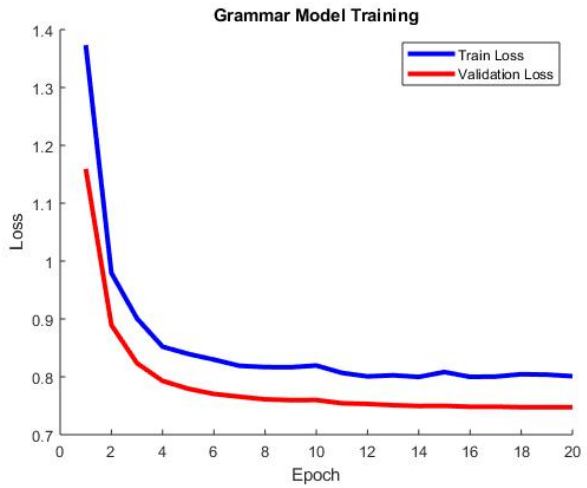


Figure 4. Cross entropy loss loss vs epoch

parts of speech are able to capture more variations. The hyper parameter and for tuning both grammar and language model is shown in the table below:

Table 4. Hyperparameters of AWD-LSTM Model

| | Language Model | Grammar Model |
|---|---|---|
| Batch size | 64 | 64 |
| Learning rate | 0.07 | 0.031 |
| Perplexity | 3.56 | 2.16 |

## 5. Beam Search

### 5.1. Beam Search of Selecting Grammar Rules

Beam search generates the top k successors with the highest probability at each level of the tree. It then sorts them based on probability and keeps only the top number of nodes. The maximum number of nodes it keep is called the beam width. Given a fixed length of words, A beam search with a beam width of 160 is applied to search for the sentence grammar rules that have all the parts of speech of the keywords. Finally, at the last level of the tree, the top 32 grammar rules are selected.

### 5.2. Beam Search of Finding Sentences

As shown in figure 5, given a grammar rule that defines the parts of speech for each word in the sentence, we only need to search the word of a particular parts of speech rather than from the entire vocabulary, which can be described as the following equation:

$$w_i = \underset{w \in \mathcal{V}_t}{\operatorname{argmax}} P(w|w_1, \ldots, w_{i-1}) \tag{9}$$

where $w_i$ is the $i_{th}$ word. $\mathcal{V}_t$ is the set of words of a particular parts of speech tag. Since the parts of speech is given before, it is independent from the previous words. So the log-likelihood of a sentence given the parts of speech $logP_T(s)$ can be expressed the addition of the log-likelihood of a sentence and the log-likelihood of a word given a part-of-speech tag:

$$logP_T(s) = logP(s) + \sum_i (logP(w_i|t_i)) \tag{10}$$

where $t_i$ is the $i_{th}$ part-of-speech tag, $P(s)$ is defined as the sentence probability without part-of-speech tag as shown in equation 2. $P(w_i|t_i)$ is the probability of word $w_i$ given a part-of-speech tag $t_i$. Compared this equation with equation 2, we can find that the probability of a sentence now heavily depends on if it has the right parts of speech, which are is from a a grammar model with much lower perplexity. Therefore, this helps the model to generate grammatically correct sentences.

To further refine the search, branches with repetitive keywords are removed immediately for each level of the tree before it reaches the leaves level. A beam width of 1600 is chosen for this search, so at the leaves level, 1600 sentences are generated. Assume that there are 32 grammar rules that have all the keywords part-of-speech tags, a total of 1600x32 = 51200 sentences are generated for each sentence length (number of words in a sentence). Then it searches across different sentence lengths. Finally, from the sentences that can fit all keywords given those grammar rules, we select the one that has lowest perplexity.

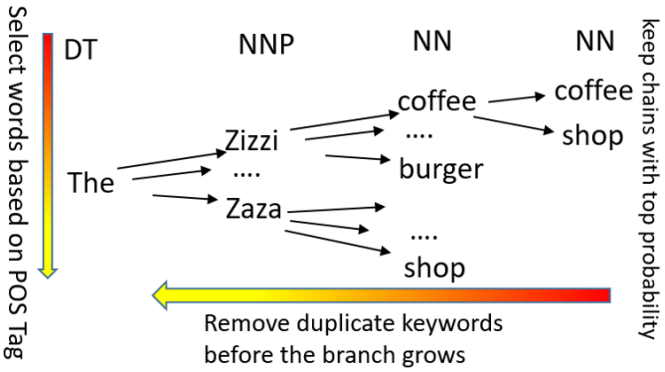


Figure 5. Beam search: search from only the words that have the right POS

Table 5. Hyperparameters of Beam Search

| | Sentence | POS |
|---|---|---|
| Beam size | 1600 | 160 |
| Successors | 10 | 4 |

Note the successors here are the maximum number of words or parts of speech it will keep.

## 6. Evaluation

4000 texts with less than 6 keywords are randomly selected as the test set, which is not used during training. The number in the keywords are all replaced with a fixed number. The model is able to generate grammatically correct sentence given the keywords, as shown in table 6 below. However, sometimes it adds extra information. For example, given the keywords Strada restaurant, the average, it adds extra information that it is a family friendly restaurant. For some of them, it can misinterpret the meaning of keywords such as high customer rating instead of high price in the third example.

Table 6. Sample generated sentences

| Keywords | Reference | Generate |
|---|---|---|
| Strada restaurant average | Strada is a restaurant with an average customer rating | a strada is a family friendly restaurant that has average food. |
| Aromi restaurant Chinese riverside | Aromi is a restaurant which provides Chinese food in the riverside area | riverside aromi is a low customer restaurant that serves chinese customer. |
| The Punter English high | The Punter is a restaurant with high prices | the punter is english price with a high customer rating. |

*Note: many of the sentences has the structure as "[restaurant name] [is]". This makes the generated sentence has a similar structure and such task easier for generating a template given label data. In the future, a more variety of sentences structure should be tested.

## 7. Comparison

As shown in table 7, comparison with other work shows that the BLEU score is low mainly because the model sometimes adds extra information. It measures the precision, which is the ratio of the number of overlapping words to the number of words in the generated system. So the more extra information, the lower the score it has. The ROUGE-l measures the longest common subsequence. In this case, the ROUGE-l score on recall shows competitive performance with other published work, as the recall is the ratio of the number of overlapping words to the number of words in the reference. So the impact of adding extra information in the generated sentence does not affect its score as much as BLEU.

Table 7. Comparison with other work on E2E dataset

| | Rouge-L | BLEU | Method |
|---|---|---|---|
| This work | 0.56 | 0.31 | Unsupervised |
| TGEN[7] | 0.68 | 0.65 | Seq2seq |
| FORGE3[6] | 0.56 | 0.46 | Templates |
| Sheff1[1] | 0.67 | 0.6 | Imitation learning |

## 8. Conclusion

This paper proposed a dual-models approach for natural language generation of the task-oriented dialogue system. It demonstrates that it is possible to use label-free text data to generate grammatically correct sentences given keywords and that a general language model can be applied to train a grammar model consisted of parts of speech and obtain low perplexity rating. The high ROUGE-l score on recall shows the success of the algorithm. The paper also proposed a modified beam search for the model. The source code can be found at https://github.com/dianhuang/proj.

## 9. Future Work

The grammar model achieves very low perplexity, so it is possible to train it on a more general corpus so that the sentence generated can have more variation in structure. The generated sentences sometimes include extra information not present in the keywords. A method such as finding the shortest sentence that includes all keywords may be able to resolve this problem. The generated response can also be more conversational by training on dialogue data.

## 10. Contribution

The author would like to thank Snehasish Mukherjee for showing how a dialogue system works and the general approach to natural language generation.

The author of the paper did all the above work independently.

## References

[1] M. Chen, G. Lampouras, and A. Vlachos, "Sheffield at e 2 e : structured prediction approaches to end-to-end language generation," 2018.

[2] O. Dusek and F. Jurcícek, "A context-aware natural language generator for dialogue systems," in *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, 2016, pp. 185–190. [Online]. Available: http://aclweb.org/anthology/W/W16/W16-3622.pdf

[3] K. Kumagai, I. Kobayashi, D. Mochihashi, H. Asoh, T. Nakamura, and T. Nagai, "Human-like natural language generation using monte carlo tree search," in *Proceedings of the INLG 2016 Workshop on Computational Creativity in Natural Language Generation, CC-NLG 2016, Edinburgh, UK, September 2016*, 2016, pp. 11–18. [Online]. Available: https://doi.org/10.18653/v1/W16-5502

[4] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," *CoRR*, vol. abs/1606.01541, 2016. [Online]. Available: http://arxiv.org/abs/1606.01541

[5] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models," *CoRR*, vol. abs/1708.02182, 2017. [Online]. Available: http://arxiv.org/abs/1708.02182

[6] S. Mille and S. Dasiopoulou, "Forge at e2e 2017," *E2E NLG Challenge System Descriptions*, 2018. [Online]. Available: http://www.macs.hw.ac.uk/InteractionLab/E2E/final_papers/E2E-FORGe.pdf

[7] J. Novikova, O. Dusek, and V. Rieser, "The E2E dataset: New challenges for end-to-end generation," *CoRR*, vol. abs/1706.09254, 2017. [Online]. Available: http://arxiv.org/abs/1706.09254

[8] Y. Puzikov and I. Gurevych, "E2E NLG challenge: Neural models vs. templates," in *Proceedings of the 11th International Conference on Natural Language Generation*. Tilburg University, The Netherlands: Association for Computational Linguistics, Nov. 2018, pp. 463–471. [Online]. Available: https://www.aclweb.org/anthology/W18-6557

[9] S. Sharma, J. He, K. Suleman, H. Schulz, and P. Bachman, "Natural language generation in dialogue using lexicalized and delexicalized data," *CoRR*, vol. abs/1606.03632, 2016. [Online]. Available: http://arxiv.org/abs/1606.03632

[10] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language TechnologyNAACL 03*, vol. 1, 03 2004.