Does Curriculum Learning help Deep Learning for Natural Language Generation?

Sandhya Singh[1], Kevin Patel[3], Pushpak Bhattacharyya[3],
Krishnanjan Bhattacharjee[2], Hemant Darbari[2], Seema Verma[1]

[1]Banasthali Vidyapith, Rajasthan, [2]C-DAC, Pune, [3]CFILT, IIT Bombay
sandhya.singh@gmail.com, {kevin.patel,pb}@cse.iitb.ac.in,
{krishnanjanb,darbari}@cdac.in, seemaverma3@yahoo.com

Abstract

Natural Language Generation (NLG) is a challenging problem in the field of Artificial Intelligence. The difficulty stems from the natural language's flexibility to convey the same message in different ways. Psycholinguists have always believed that learning simpler sentences early can lead to complex sentence creation using the same knowledge.This is also the intuition behind curriculum learning. Thus, in this paper, we investigate the use of curriculum learning for natural language generation. We show that curriculum learning is a promising training methodology for deep learning systems for NLG. We show this by reporting improvements obtained using i) a particular curriculum strategy and ii) augmenting data using curriculum logic. We use TGen, a deep learning based NLG system, for experimentation. We use 5 metrics for NLG evaluation and 8 metrics for readability evaluation. Our quantitative and qualitative evaluation shows that the system trained using curriculum methodology produces better quality text as compared to the system trained on normal data.

1 Introduction

Natural language generation involves creating a natural language(NL) sentence from a non-linguistic input, which can be a structured meaning representation (MR), statistical data or a parse tree structure. An example of MR to natural language is *[name(Barbeque Nation), eatType(restaurant)]* → *Barbeque Nation is a restaurant.* Ever since the pioneering work by (De Smedt et al., 1996; Reiter and Dale, 2000), this has been an active area of research (Gatt and Reiter, 2009; Lampouras and Androutsopoulos, 2013; Kondadadi et al., 2013; Wen et al., 2015b,a).

A major hurdle in applying traditional machine learning techniques for NLG is the possibility of having more than one correct sentences for a given meaning representation. In machine learning terminology, this implies two things : First, there can be more than one correct label for a given input. Second, the set of possible labels is infinite. Thus, computing the loss function, which is a standard component in many machine learning algorithms, is difficult (Lampouras and Androutsopoulos, 2013). Therefore, most approaches default to template based learning or guided learning mechanisms. The sentences generated using such approaches feel artificial, as there is not much variation in the generated sentences (Langkilde and Knight, 1998; Deemter et al., 2005; Manurung et al., 2008).

Deep learning has made tremendous strides in different areas of Natural Language Processing. Much of its success can be attributed to two factors: its ability to lend itself to representation learning and the emergence of set of techniques which make effective training of deep neural networks possible. Deep learning has been successfully applied in several NLP tasks : Part of Speech Tagging (Collobert and Weston, 2008), Sentence Classification (Kim, 2014), Sentiment Analysis (Liu et al., 2015), Sarcasm Detection (Joshi et al., 2016). Recently, Wen et al. (2015b); Dušek and Jurcicek (2016) *etc.* have investigated the use of deep neural networks to generate natural language.

Humans learn better through an organized step-wise manner, exploiting already learned concepts while learning new difficult concepts. Tailoring training data in such a manner to assist a machine learning system is known as curriculum learning (Bengio et al., 2009; Fan et al., 2017).

We intuit that an NLG system may learn to form complex sentences by leveraging knowledge of forming simple sentences. Thus, in this paper, we raise the following question:

Does curriculum learning help improve the output quality and performance of deep learning based Natural Language Generation?

We investigate this question using TGen, a sequence to sequence based natural language generator (Dušek and Jurcicek, 2016). We performed a quantitative evaluation of the generated text. We also qualitatively evaluate coverage of MRs and ambiguity factor of the generated text. Our preliminary evaluations provide the following evidence for a positive reply to the above question:

- System trained using a length based curriculum strategy performs better than system trained using randomly shuffled data.

- System trained using curriculum-augmented data performs better than system trained on original data.

The rest of the paper is organized as follows: Section 2 describes related work to the problem. Section 3 provides the experimental setup of our evaluation. Section 4 discusses our quantitative and qualitative analysis followed by conclusion and future work.

2 Related work

2.1 NLG and Deep Learning

Chang et al. (2015) experimented with deep neural networks for sentence generation as well as other related features for generation. Wen et al. (2015a) used a joint recurrent and convolutional neural network for dialogue generation. Lampouras and Vlachos (2016) developed Locally Optimal Learning to Search (LOLS) framework, which used imitation learning to generate sentences. Wen et al. (2015b) proposed a semantically conditioned Long Short-term Memory(LSTM) for language generation. It was trained to learn from unaligned data.

Finally, Dušek and Jurcicek (2016) proposed TGen, a sequence to sequence based encoder-decoder architecture along with beam search and a reranker to generate natural language sentence from meaning representation. The architecture combines sentence planning and surface realization stages of generation and produces strings using LSTM based sequence generator.

These developments have led to a state where current NLG systems are able to generate more natural and varied output in comparison to earlier rule and template based generation. However, extremely large output space still leaves a lot scope for improvement.

2.2 Curriculum Learning, Deep Learning and NLP

The training criteria in most deep neural networks is non-convex. This adds two extra challenges to the problem of learning: the quality of the local minima obtained, and the speed of convergence towards that minima. Bengio et al. (2009) showed that curriculum learning addresses both these challenges positively. They demonstrated the effectiveness of curriculum learning for language modeling, among other tasks. This effectiveness was soon exploited by others. Shi et al. (2015) experimented with RNN language model for *within-domain adaptation* and *limited data within domain adaptation* using curriculum learning with improved outcomes. Cirik et al. (2016) studied the performance of curriculum learning on long-short term memory networks for sentiment analysis task. Similarly, Sachan and Xing (2016) showed that data ordering of simple hand crafted questions improved performance of question answering using deep neural networks. Liu et al. (2018) also experimented with curriculum learning approach for natural answer generation. This motivated us to explore NLG using curriculum learning.

Meaning Representation	name[Alimentum], area[city centre], familyFriendly[no]
Natural Language	Alimentum is not a family-friendly arena and is located in the city centre.

Table 1: Sample of input MR-NL utterance pair used for training

3 Experimental Setup

3.1 Data

3.1.1 Original Data

We used E2E-challenge dataset (Novikova et al., 2017). It is from the restaurant domain with around 42K sentences in the form of dialogue act-based meaning representations(MRs) coupled with its natural language utterances. The natural language text of the MRs from the dataset show open vocabulary with complex sentence structures and varied discourse patterns. Thus we conclude that this dataset is really good representative of the real world NLG problem. A sample MR and its NL utterance is shown in Table 1.

3.1.2 Creating curriculum-augmented data

Meaning Representation (MR)	Natural Language (NL)
name[Fitzbillies]	Fitzbillies is a restaurant .
name[Fitzbillies], eatType[coffeee shop]	Fitzbillies is a coffee shop .
name[Fitzbillies], food[French]	Fitzbillies serves french food .
name[Fitzbillies], area[riverside]	Fitzbillies is located in riverside .
name[Fitzbillies], eatType[coffeee shop], food[French]	Fitzbillies is a coffee shop serving french cuisine .
name[Fitzbillies], eatType[coffeee shop], area[riverside]	In the riverside area is a coffee shop named Fitzbillies .
name[Fitzbillies], food[French], area[riverside]	Fitzbillies serves french food in riverside area .
name[Fitzbillies], eatType[coffeee shop], food[French],area[riverside]	Fitzbillies is a coffee shop serving french food in riverside area.

Table 2: Sample of Curriculum data created from training MR: *”name[Fitzbillies], eatType[coffeee shop], food[French], area[riverside]”*

We create the curriculum-augmented data as follows. Let the original training set be $S_{original}$. Let the TGen model trained on randomly shuffled $S_{original}$ be $TGen_{original}$. For each MR of $S_{original}$, a set S_{comb} of all possible MR combinations with name field as constant factor is created. From this set S_{comb}, a subset S_{uniq_comb} of unique MR combinations is created. This fills the MR field of the dataset. To create the corresponding NL utterances, we first pass S_{uniq_comb} to $TGen_{original}$. The automatically generated utterances are then manually verified (for missing phrases corresponding to MR tag) and added to S_{uniq_comb}. This fills the NL field of the dataset.

We then remove duplicates from the combined set $S_{original} + S_{uniq\_comb}$ to create our training set $S_{augmented}$. The size of this $S_{augmented}$ dataset is 75K sentences. A sample of curriculum data is shown in Table 2.

3.2 Ordering Strategy

As discussed earlier, a major part of effectiveness of curriculum learning comes from the way data is ordered. We want the system to first learn to form simple sentences, and then move on to learning to form complex sentences. We define simplicity of a sentence in terms of the length of MR+NL. Thus in our experiments, we use the following ordering strategies:

- **Shuffled**: In this strategy, the data is randomly shuffled. This strategy is a baseline for comparison.
- **Curriculum**: In this strategy, the data is sorted based on the length of MR + NL.

3.3 Training

The training data was preprocessed to delexicalize the *name* and *near* MR slots to reduce the data sparsity, as recommended by Dušek and Jurcicek (2016). A sequence representation of each MR sequence is then created by joining the triplet information containing type, slot name and value for each MR slot and converted to a vector representation. We chose the string mode of TGen where it combines sentence planning and surface realization stages of NLG architecture(Konstas and Lapata, 2013). The TGen generator model was trained using a LSTM based seq2seq encoder-decoder architecture with 128 hidden units, embedding size 50, learning rate as 0.001 and batch size 20 along with Adam optimizer (Kingma and Ba, 2014). A reranker with beam size 10 is used for generation(Bahdanau et al., 2014).

We trained four different models as follows:

- **Sys1**: TGen trained on original data ($S_{original}$) with shuffled ordering strategy.
- **Sys2**: TGen trained on original data ($S_{original}$) with curriculum ordering strategy.
- **Sys3**: TGen trained on curriculum-augmented data ($S_{augmented}$) with shuffled

ordering strategy.

- **Sys4**: TGen trained on curriculum-augmented data ($S_{augmented}$) with curriculum ordering strategy.

3.4 Evaluation Metrics

We evaluated the quality (adequacy and fluency) of generated text using automatic evaluation metrics which measure the word-overlap with respect to the reference sentences - BLEU(Papineni et al., 2002), NIST(Doddington, 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004) and CIDEr (Vedantam et al., 2015) scores (as per the E2ENLG challenge). These metrics capture the degree of intended content that is transferred via the generated text.

We also evaluated the readability of the generated text using the automatic evaluation metrics which are:

1. The Flesch Reading Ease formula (FRE) (Flesch, 1948) that calculates the reading level of the content. It is scored from 1-100 with 100 being the easiest to comprehend and 1 being the hardest and confusing. A score of 60-70 is preferred for standard content.
2. The Flesch-Kincaid Grade Level (FKG) (Kincaid et al., 1975) score captures the level of content in the form of grade from 0-12 as per the standard accepted globally.
3. SMOG Index (SMOG) (Mc Laughlin, 1969) suggests the years of education needed to understand the piece of writing. The scores range from 5-18.
4. Gunning FOG Formula (GFOG) (Gunning, 1969) estimates the years of formal education needed to understand the text on the first reading. It ranges from 6-18.
5. Automated Readability Index (ARI) (Kincaid et al., 1975) is devised to gauge the understandability of a text. The scores grade the level needed to comprehend the text and varies from 1-12.
6. The Coleman-Liau Index (CLI) (Coleman and Liau, 1975) estimates the years of formal education required to understand the text on first reading with scores ranging from 1-12.
7. Linsear Write Formula (LWF) calculates the readability based on sentence length and no. of words with more syllables.
8. Dale-Chall Readability Score (DCRS) (Chall and Dale, 1995) measures the comprehension difficulty while reading the text.

One can infer from above that the major factors affecting these metrics are complexity of the vocabulary and average sentence length.

4 Results and Analysis

4.1 Quantitative Analysis

Table 3 shows the adequacy and fluency scores of the generated text. One can observe that, with respect to a particular dataset, the system trained using curriculum as ordering strategy is performing better than the system trained using shuffled ordering strategy **(Sys2 > Sys1 and Sys4 > Sys3)**. One can also observe that, with respect to a particular ordering strategy, the system trained using curriculum-augmented data is performing better than the system trained on original data **(Sys3 > Sys1 and Sys4 > Sys2)**. Overall, the system trained on curriculum-augmented data with curriculum as ordering strategy is performing the best.

Dataset	Original		Curr-aug	
Ordering	Shuf	Curr	Shuf	Curr
	Sys1	**Sys2**	**Sys3**	**Sys4**
BLEU	0.60	0.61	0.61	**0.64**
NIST	7.90	7.84	8.07	**8.39**
METEOR	0.41	0.42	0.41	**0.45**
ROUGE_L	0.66	0.67	0.66	**0.69**
CIDEr	1.95	2.00	2.00	**2.23**

Table 3: Results of adequacy and fluency evaluation. {original, curr-aug} indicates whether the training data was original or augmented. {Shuf, Curr} indicates the ordering strategy. Each value is the average of three runs. Each jump is statistically significant(p-values < 0.01)[†]

Next, we proceeded to measure the gains obtained using curriculum techniques. In this regard, we perform the remaining comparisons between the system without any curriculum learning component **(Sys1)** against the system with both curriculum learning components **(Sys4)**.

We observed that **Sys4's** percentage of MR tag coverage in the generated text is marginally higher (by 0.5953 percent points) than **Sys1's** coverage. Another observation

Metric	Sys1	Sys4	Ref
FRE	**71.80**	74.78	67.15
FKG	**6.43**	6.24	7.73
SMOG	6.9	**18.7**	11.6
GFOG	16.27	**16.45**	17.90
ARI	6.30	**6.44**	8.55
CLI	**8.00**	7.57	9.41
LWF	5.24	**5.77**	6.51
DCRS	7.85	7.85	8.23

Table 4: Results of readability evaluation. Except FRE, the higher the score the better

was that the average sentence length of text generated using **Sys4** was greater the average sentence length of text generated using **Sys1** by 2.36 percent points.

This shows that quantitatively, curriculum learning techniques can be helpful for deep learning based NLG.

Table 4 shows the readability evaluation of the generated text. The scores indicate that **Sys4** is generating relatively complex sentences as compared to **Sys1**.

4.2 Qualitative Analysis

Now we highlight some of the aspects where curriculum learning helped text generation.

MR	name[The Waterman], food[Fast food], priceRange[moderate], customer rating[3 out of 5], area[riverside], kidsFriendly[yes]
Sys1 Output	The Waterman is a kid friendly fast food restaurant with a moderate price range. It has a customer rating of 3 out of 5.
Sys4 Output	The Waterman is a kid friendly fast food restaurant in the riverside area with a moderate price range and a customer rating of 3 out of 5.

Table 5: Example showing how **Sys4** does handle the tag *area* which is dropped by **Sys1**

Consider the example in table 5. Here, it is evident that **Sys1** did not cover the *area[riverside]* term of the MR. Whereas, **Sys4** was able to accommodate it correctly.

Now, consider the example in table 6. Training data for both models included the Training MR and Training Reference, which had *food[French]* and *'is a French Pub'*. Now, the test MR has *food[English]*. One may observe that the **Sys1** generated a phrase similar to the reference, *i.e. 'is a English Pub'*. Note that this is a slightly ambiguous phrase, and can mean *a pub serving English food* or *a pub managed by English people*. Whereas, **Sys4** is able to generate a relatively unambiguous *'is a pub providing English food'* phrase. **Sys1** also has an ambiguous anaphora *'It'* (could mean both *The Plough* or *Cafe Rouge*), which is not the case with **Sys4**. Thus **Sys4** is generating better text here.

Training MR	name[The Plough], eatType[pub], food[French], priceRange[moderate], kidsFriendly[no], near[Cafe Rouge]
Training NL	The Plough is a French pub, which is not kid friendly. The price range is moderate and is located near caffe Rouge.
Test MR	name[The Plough], eatType[pub], food[English], priceRange[more than £30], children-friendly[yes], near[Cafe Rouge]
Sys1 Output of Test MR	The Plough is an english pub near Cafe Rouge. It is child friendly and has a price range of more than £30.
Sys4 Output of Test MR	The Plough is a pub providing english food in the high price range. It is located near Cafe Rouge and is children friendly.

Table 6: Example demonstrating how ***Sys4*** generates relatively unambiguous text

Using these examples, we qualitatively argue that curriculum system is better at NLG.

5 Conclusion and Future Work

In this paper, we proposed using curriculum learning as a training methodology for deep learning based natural language generation systems. We argued that both curriculum ordering strategy and curriculum augmented data could help learning natural language generation. Our quantitative evaluation showed that text generated using a system trained with either curriculum ordering strategy or curriculum-augmented data or both was better in terms of both adequacy and fluency, as well as readability when compared to a system trained on randomly shuffled original data. This was established via a set of different evaluation metrics. Our qualitative evaluation indicates that using curriculum led to better coverage and less ambiguity. Thus we conclude that curriculum learning based training methodology is indeed a promising methodology for deep learning based NLG systems. In the future, we will investigate MR based and vocabulary based approaches for designing curriculum strategies and data-augmentation for natural language generation.

[†] We used Welch unpaired t test for significance testing.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume III, and John Langford. 2015. Learning to search better than your teacher.

Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML)*, volume 307.0, pages 160–167. ACM.

Koenraad De Smedt, Helmut Horacek, and Michael Zock. 1996. Architectures for natural language generation: Problems and perspectives. In *Trends in Natural Language Generation An Artificial Intelligence Perspective*, pages 17–46. Springer.

Kees Van Deemter, Mariët Theune, and Emiel Krahmer. 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Ondřej Dušek and Filip Jurcicek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics.

Yang Fan, Fei Tian, Tao Qin, Jiang Bian, and Tie-Yan Liu. 2017. Learning what data to learn. *arXiv preprint arXiv:1702.08635*.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Albert Gatt and Ehud Reiter. 2009. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 90–93. Association for Computational Linguistics.

Robert Gunning. 1969. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13.

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1006–1011. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical nlg framework for aggregated planning and realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1406–1415.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *Journal of Artificial Intelligence Research*, 48:305–346.

Gerasimos Lampouras and Ion Androutsopoulos. 2013. Using integer linear programming for content selection, lexicalization, and aggregation to produce compact texts from owl ontologies. In

Proceedings of the 14th European Workshop on Natural Language Generation, pages 51–60.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1101–1112.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum learning for natural answer generation. In *IJCAI*, pages 4223–4229.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443. Association for Computational Linguistics.

Ruli Manurung, Graeme Ritchie, Helen Pain, Annalu Waller, Dave O'Mara, and Rolf Black. 2008. The construction of a pun generator for language skills development. *Applied Artificial Intelligence*, 22(9):841–869.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 453–463.

Yangyang Shi, Martha Larson, and Catholijn M Jonker. 2015. Recurrent neural network language model adaptation with curriculum learning. *Computer Speech & Language*, 33(1):136–154.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015b. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.