

A Comparison of Data-Driven and Template-Based Approaches to Natural Language Generation

Jarryd Dunn
University of Cape Town
dnnjar001@myuct.ac.za

ABSTRACT

Data-driven and template-based systems are two popular approaches to developing Natural Language Generation (NLG) systems. There are various problems associated with the texts produced by both these methods, which raises the question of whether one approach yields better results than the other. To answer this question a data-driven as well as a template-based system was developed. The data-driven system consisted of separate sentence planner and linguistic realisation modules. These systems then generated eight utterances for the same set of input Meaning Representations (MRs). These input MRs came from the Wikipedia Person data set. The utterances generated by the systems were then evaluated by 91 human judges. The human judges were presented with pairs of texts generated using different methods and asked to rate each text based on its clarity and fluency. The judges were then asked to choose which text was more natural. The texts in the questionnaire included reference texts corresponding to the input MR to act as a base line.

The template-based system was found to produce texts that were significantly more fluent than both the data-driven and reference texts. The template-based system also produced texts with higher clarity ratings than the data-driven system and text of similar clarity to the reference texts. Despite this, the reference texts were still selected as being more natural than those produced by the template-based system. While the text produced by the data-driven systems were judged to be less natural than both the template-based system and reference texts. This is most likely due to the semantic errors which made it obvious that a human did not write these texts. Thus we conclude that the template-based approach may provide a more promising approach to constricting a NLG system for the data set than the data-driven system.

KEYWORDS

Natural Language Generation, NLG, Data-to-text Generation, Machine Learning, Data-driven Natural Language Generation, Template-based Natural Language Generation

1 INTRODUCTION

Data-to-text NLG refers to the process of producing a natural language utterance from some non-linguistic representation (the input Meaning Representation (MR)) [13]. For the NLG systems developed a meaning representation consists of one or more slot-types (tokens) each corresponding to one or more slot-values (token values). For example, an MR may have the slot-type, slot-value pairs shown in Table 1. Given this MR a data to text NLG system would try to produce a natural language utterance such as “Alice (born 1970/01/01) is interested in cryptography and encryption.”

Slot-Type	Slot-Value
Name	Alice
Date of Birth	1970/01/01
Interest	Cryptography
Interest	Encryption

Table 1: Slot-type and slot-value pair example

There are several approaches to creating a data-to-text NLG system. More recently data-driven approaches to data-to-text NLG have become popular. Data-driven approaches tend to produce utterances with more variation and a more natural style and tone [11, 18]. Unfortunately, these utterances are also prone to syntactic and grammatical errors [2] as well as struggling with ordering, omitting, repeating and hallucinating facts [9]. Hallucinating occurs when slot-types that are not in the MR appear in the generated text. Another option is to take a template-based approach. A template-based approach makes use of one or more templates to produce a natural language utterance. These templates may either be handcrafted or learnt from a data set [12] using machine learning techniques. The natural language utterance is realised by inserting slot-values into the template, based on their slot-types. Template-based systems tend to produce grammatically correct utterances since the creator often has a large amount of control over the structure of the final template; however, this approach may not scale well to large and varied data sets since template-based approaches rely on redundancy to produce appropriate templates [8, 17].

When compared to a template-based approach, using a data-driven approach can decrease the amount of time it takes to develop an NLG system [2, 12]. However this also means that there is less control over the utterances produced by data-driven Natural Language Generation (NLG) systems. This means that the systems have the potential to produce output that is less fluent or understandable, due to semantic and grammatical errors, than the outputs of a template-based system [2]. Although the utterances produced by the data-driven system may appear more natural and varied. Some of these semantic errors may be removed by applying additional rules after generating the utterance [12].

One of the main goals of an NLG system is to make data more understandable. Therefore, it is desirable for the system to produce utterances that are indistinguishable from human-written texts. This leaves us with the question of whether humans prefer utterances generated by a data-driven NLG system, which may contain semantic and grammatical errors, to those generated by a template-based system. Although the utterances produced by the template-based system may be less natural and display less variation.

Part of the difficulty with evaluating the utterances produced is in how to determine if one piece of text is better than another. The quality of the utterance can be defined based on numerous factors such as its fluency, clarity, grammatical correctness, semantic correctness and naturalness. In this context, clarity refers to how understandable and clear the utterance is [15] and fluency refers to how easy the utterance is to read [15]. Likewise, for this paper, naturalness refers to how closely the text resembles natural human writing styles and nuances. For this paper, the quality of a text was judged by humans based on its clarity, fluency and naturalness.

This paper focuses on the construction and performance of the data-driven NLG system to be compared with a template-based system. This system is split into two main modules: the sentence planner and linguistic realiser. The sentence planning module is responsible for determining how the tokens from the input MR should be split into sentences and the order of these sentences. This results in a sentence plan. The linguistic realiser takes in the sentence plan and renders it as a natural language utterance.

The utterances produced by the data-driven and template-based systems were then compared using human judges. This was done to determine which system produced utterances that humans found to be more natural.

Section 2 looks at previous work comparing data-driven and template-based NLG approaches. Sections 4 and 5 examine the sentence planner and linguistic realiser of data-driven system. In Section 3 the data set used and the preprocessing of the data set is described. The template-based system that the data-driven system is compared against is briefly described in Section 6. The experiment comparing the data-driven and template-based systems is outlined in Section 7. Section 7 also includes several minor experiments on the data-driven system to determine the effects of different mechanisms on the results produced. The results of the comparison and performance experiments are shown in Section 8. Section 9 discusses some aspects of the results. Finally, Section 10 contains the conclusion.

2 RELATED WORKS

Recent NLG systems include those produced by Moryossef et al. [9], Lampouras and Vlachos [6] and Dušek and Jurčiček [1].

Moryossef et al. [9] split the system into planning and realisation modules. The planning phase generates a sentence plan that is then used by the realisation module to generate the natural language utterances. Using the two module approach led to improvements in the BLEU scores and manual evaluations of the utterances produced.

Lampouras and Vlachos [6] developed a non-neural approach by viewing NLG as a classification problem over a delexicalised corpus. Words related to a slot-type from the input MR were grouped together. Words from this group would then be used to express this slot-type to generate the natural language utterance.

The TGEN system developed by Dušek and Jurčiček [1] is an end-to-end NLG system. An encoder-decoder module was used to learn to map the input MR to an abstract representation of the sentences that will form the natural language utterances. An external rule-based surface realiser is then used to produce the final natural language sentence. The use of the rule-based surface realiser does mean that the system may not perform well on other data sets [6].

Puzikov and Gurevych [12] have previously compared neural (data-driven) models and template-based approaches for Natural Language Generation. They constructed a data-driven system based on TGEN [1] for the E2E NLG Challenge (D-model) as well as a template-based model (T-model). Automatic evaluation metrics were then used to compare the texts produced by the two models. Puzikov and Gurevych determined that the additional costs of generating complex data-driven systems, in terms of both the time and computing resources, may not be justified. Thus, it may be better to take a template-based approach to NLG.

The main difference between the comparisons is the data set used. The Wikipedia Person data set contains MRs with a variable number of unique slot-types. In general, each MR contained more slot-values than the data set that was used in the E2E NLG Challenge. The reference sentences also tended to be longer and more varied in the Wikipedia Person data set. Furthermore, Puzikov and Gurevych [12] used automated evaluation metrics to evaluate the utterances produced by their systems; however, there is only a weak correlation between automated evaluation metrics and human judgement [10].

3 DATA SET

The Wikipedia person data set [16] was used to train and test the NLG system. This data set contains almost half a million entries collected from Wikipedia pages describing people. The slot-types and slot-values in the MR are collected from the Wikipedia information boxes. The reference text associated with each MR consists of sentences from the Wikipedia page that contain one or more of the slot-values, as shown in appendix A. The entries from the data set were randomly distributed between three subsets (training, validation and testing) with 60% of the entries in the training data set, 30% in the validation data set and 10% in the testing data set. A summary of the number of reference sentences, slot-types and slot-values are shown in Table 2. The training data set was used to train models. The validation data set provided the input MRs for testing the linguistic realiser and sentence planner. Finally, the testing data set was used to provide the input MRs to be compared with the template-based system. This data set poses several interesting challenges since the reference text is made up of multiple sentences (mean of 9.11 sentences per reference text¹). These sentences also tend to have relatively few token values per sentence (mean of 1.19 token values per sentence¹). Thus many sentences contain only one token value and the reference sentences tend to include a significant amount of detail that doesn't relate to the token value. This makes it less probable for the encoder-decoder module, used in the linguistic realiser, to learn a mapping from a sequence of tokens to a natural language sequence.

3.1 Preprocessing

To try and improve the quality of the sentence plans and natural language utterances generated by the NLG system several processes were carried out over the data set to delexicalise the data set. The delexicalisation process involves extracting tokens and token sentences. Token sentences are formed by removing all non-token words from the reference sentences (Appendix A Table 8 shows the

¹ For the first 1000 entries in the validation subset

² Only references including one or more token values are included

Subset	Entries	Reference Sentences ²	Token Types	Token Values
Training	257248	2203014	1029	5471715
Validation	128625	1107768	821	2750577
Testing	42875	116149	558	914818
Total	428748	3426931	1265	9137110

Table 2: Wikipedia Person Data Set Metrics

delexicalised sentences produced from the reference texts in Table 7).

The reference texts in the data set were delexicalised by replacing slot-values with their corresponding slot-types for each entry in the data set. An interesting feature of this data set is that some slot-values include slot-types as qualifiers. These values are delexicalised by concatenating the parent slot-type and slot-token type, see the *Goals* slot-type in appendix A Table 7. This allows the delexicalised value to be more specific than the qualifier token on its own which should allow for better sentence planning and linguistic realisation.

Once the data set has been delexicalised it can be used to produce the data required to train the sentence planner and linguistic realiser. The sentence planner requires that each set of reference texts is split into token sentences where the “<end>” token is used to denote the end of a sentence. These token sentences can be used to learn which token is most likely to come next given a sequence of tokens. The training data for the linguistic realiser is created by matching each token sentence with the corresponding delexicalised reference sentence. These are then written to separate files to form the inputs and labels for the encoder-decoder model.

Since some sentences contain no slot-values or only a single slot-value a cutoff mechanism was added to the delexicalising process. The slot-values cutoff is the minimum number of slot-values that a sentence needs to contain in order to be used for training the NLG system. This was done to try and improve the quality of the surface realiser by reducing the number of reference sentences corresponding to a sequence of tokens.

4 SENTENCE PLANNER

The sentence planner forms the first part of the NLG system. During this stage, the tokens are divided into groups which will be realised into separate sentences during the linguistic realisation phase. These tokens are made up of the slot-values from the input MR as well as an end of sentence token “<end>”. Two variations of the sentence planner were constructed: a greedy sentence planner (Section 4.2) and a stochastic sentence planner (Section 4.3).

4.1 Training

The sentence planner works by generating a Markov Decision Process (MDP) type structure, the token selector, where each state consists of an n-gram of tokens (sequence of n tokens e.g Name_ID <end> would be a 2-gram of tokens). The probabilities of each token following the particular n-gram of tokens are used in a similar way to the transition probabilities in an MDP. These transition probabilities were learnt using tokens extracted from sentences included in the data set. In this implementation, the n-grams consisted of

one or two tokens. A separate process is used to determine which token is most likely to be the first in an utterance.

The probability for each token starting the utterance is also recorded. These probabilities are used to determine which token to use as the initial token when generating the sentence plan.

4.2 Generating Sentence Plans

Once the token selector for the planner has been trained it can be used to generate sentence plans. Suppose that the input MR consisted of the tokens: Name_ID, date_of_birth, sport, country_of_citizenship.

First, the starting token is chosen by taking the most likely token to start an utterance based on the probabilities calculated during training. The starting token is then used with the token selector to determine the next token to be included. This is done by choosing the most probable token that is in the MR and has not yet been placed in the sentence plan. Suppose that the date_of_birth token was selected. The current sentence plan is then “Name_ID date_of_birth” and the remaining tokens from the input MR not yet in the sentence plan are sport and country_of_citizenship. This process to select the next token is shown in fig 1. The sentence planner starts by checking for the next most probable token using the last two tokens that were added. If no suitable token can be determined then it considers only the last token added to the sentence plan. If there is still no suitable token then a token is chosen from the remaining set at random. The sentence plan is returned once all the tokens from the MR have been included or the maximum number of iterations have been reached. The maximum number of iterations used is twice the number of token-values in the MR since the Name_ID and <end> token may occur multiple times in the sentence plan. Where the Name_ID token specifies the name of the person and the <end> token specifies the end of a sentence. Thus the final sentence plan for the example input MR might be “Name_ID date_of_birth sport country_of_citizenship <end>”

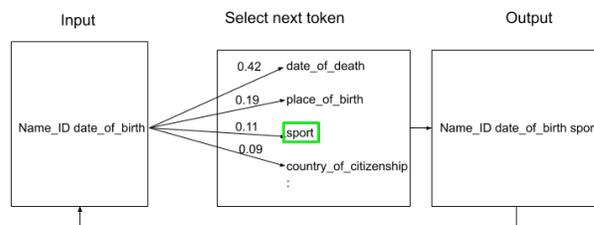


Figure 1: Sentence Planner Structure

4.3 Stochastic Token Selection

The sentence planner with stochastic token selection works in the same way as the sentence planner described above except that it was slightly modified to include non-greedy token selection. When selecting each token this method starts by generating a random number, if it is below a preset cutoff then the token is selected in the manner described above, however, if the random number is above the cutoff then a second random number is generated, the target value. The probabilities of possible next tokens are then added to a counter until the value of the counter is greater than or equal to the target value. Thus the selection is weighted by the probability of a token occurring. The most recent token is then added to the sentence plan and the method continues to select the next token or terminates if all the required tokens have been placed in the sentence plan or the maximum number of iterations have been reached. This method was implemented to allow for additional variation in the sentence plans generated while still allowing the selection of the next token to be affected by the probability of it occurring. Without this mechanism, very similar or possibly even the same sentence plan would be created every time it is given the same set of tokens.

4.4 Additional Rules

Due to the data set containing many sentences with very few tokens the number of tokens/token pairs which are most likely to be followed by an <end> token is relatively high. This resulted in the sentence planner producing many sentences consisting of just one token. To lengthen the sentences produced, and so make the utterances appear more natural, a penalty was subtracted from the probability of the <end> token occurring. This penalty value was reduced every time any other token was added to the sentence plan. When an <end> token is added the end token penalty is reset to its original value. A similar mechanism is used to control the use of the Name_ID token since sentences appear less natural when they repeatedly use a person's name rather than using referring expressions.

5 LINGUISTIC REALISER

The process the linguistic realiser uses to produce a natural language utterance from a sentence plan is outlined in figure 2. The linguistic realiser makes use of an encoder-decoder model since this model has been shown to be useful for determining the mappings between sequences [2]. The linguistic realiser uses it to learn a mapping between sequences of tokens and English utterances. The implementation used in the OpenNMT³ [5] NMTMedium model. This consists of two bidirectional Recursive Neural Networks (RNNs) using Long Short-Term Memory (LSTM) cells with attention and beam search. One acting as the encoder and the other as the decoder. The encoder takes a sentence from the sentence plan as the input and runs it through the RNN. The final hidden state is used as an encoding vector. This encoding vector is then used to initialise the initial hidden state for the decoder. The decoder then runs to produce a delexicalised English sentence. A bidirectional RNN was used since previous studies have shown them to produce more accurate results than unidirectional RNNs for applications such as

³<http://opennmt.net/OpenNMT-tf/index.html>

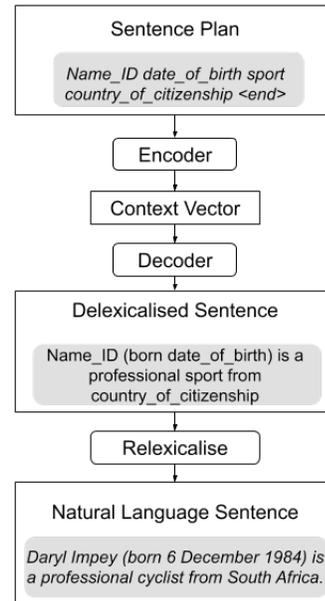


Figure 2: Encoder-Decoder model used to learn sequence mappings from a sentence plan to a natural language utterance

Neural Machine Translation (NMT) and handwriting analysis [18]. LSTM cells generally improve the ability for an RNN to capture long range dependencies between inputs than a vanilla RNN [17]. An attention mechanism is used to focus the RNN on a smaller section of the sentence this should allow the RNN to make better predictions when run on unseen inputs. Beam search is used to improve the quality of the predictions by looking at the top few most probable results (in this case three) for the predicted token rather than just a single token. This allows the system to build more than one partial hypothesis [14] with the most likely hypothesis being selected at the end as the prediction. A beam with width three was chosen as a balance between gaining the advantages of using beam search without causing too great an increase in the training time.

5.1 Training

For each sentence in the delexicalised training data set the tokens were extracted and used as the input to the encoder. The delexicalised sentence was then used as the label for the sequence of tokens. Thus the model would try and learn a mapping from the sequence of tokens to the delexicalised sentence. This created a training set of just over 2.2 million token-delexicalised sentence pairs. To improve the quality of the mappings learnt the vocabulary of the labels were reduced from just over 1.24 million words to 100,000 words. This was done by keeping the 100,000 most frequently occurring words and replacing the rest with an “<unk>” token. This should cause the model to learn more general rules for mapping the sentence plan to a natural language utterance. All the models used for the experiments were trained for 10,000 epochs.

5.2 Relexicalising Utterances

Once the model has been trained it can be used to generate delexicalised English utterances. The model is run on each sentence in the sentence plan. The results are then concatenated to form a delexicalised utterance (see figure 2). The utterance is then relexicalised by replacing the slot-types with their values from the input MR. Where there are multiple slot-values for the same MR the slot-values are selected based on the order they appeared in in the original MR. If a slot-value in the delexicalised sentence is not in the input MR then it is removed.

5.3 Post-generation Rules

In order to improve the quality of the utterances generated a few rules were applied to the generated sentences. These rules were mostly used to correct simple grammatical errors. The rules were added to ensure that:

- sentences started with capital letters and ended with some form of punctuation, though in most cases this was not necessary.
- white space was used correctly particularly around parenthesis and punctuation.
- <unk> tokens are removed.

The model was unable to learn correct pronouns. This forced a simple rule to be added to he/his with she/her if the `sex_or_gender` slot-value was female.

6 TEMPLATE-BASED SYSTEM

The template-based system was developed by Mr Matthew Poulter and follows a similar approach to van der Lee et al. [15]. It consists of three sections the document planner, micro planner and sentence realiser. The document planner selects a major template-based on the slot-types used in the input MR. The major template consists of several minor templates. The minor templates contain different ways in which the slots can be used. The minor templates are stored in a tree structure representing the overall structure of the text to form the document plan. The document plan is then passed to the microplanner. The microplanner produces a sentence plan made up of a randomly chosen minor template and the corresponding slots. The sentence plan is then passed to the SimpleNLG [3] library to be realised into a natural language utterance.

7 EVALUATION

Before coming up with the final version of the system to be used to compare the data-driven and template-based systems several configurations for both the sentence planner and linguistic realiser modules were tested. Automated evaluation metrics were used to try and gauge the performance of the system and measure any possible improvements. The measurements for these experiments were taken based on the first 1000 entries from the validation data set.

7.1 Text Evaluation via Evaluation Metrics

To evaluate the effects of alterations to the sentence planner and linguistic realiser on the utterances produced, utterances were produced for the first 1000 entries from the validation data set

and evaluated using the BLEU⁴, ROUGE⁵ and Word Error Rate (WER)⁶ evaluation metrics as well as counting the number of facts from the original MR that were missed or incorrectly added to the utterance. BLEU and ROUGE were chosen as word-overlap metrics since BLEU focuses on the precision of the generated utterance [6] while ROUGE measures the recall, precision and F1 score (see Appendix B). For both these metrics a score between zero and one is produced, where more similar texts will have a score closer to one. Both ROUGE and BLEU were used since BLEU also factors in the length of the generated and reference texts [2]. Three main forms of the ROUGE metric were used to evaluate the utterances generated. These are ROUGE-N, ROUGE-L and ROUGE-W (see Appendix B.1). Word Error Rate (WER) is a string-distance metric calculated as the number of substitutions, insertions and deletions required to turn the generated text into the reference text divided by the number of words in the reference.

7.2 Sentence Planners

To evaluate a sentence plan the recall, accuracy, precision and F1 measure⁷ (see Appendix B) of the sentence plan produced for some MR was calculated. These calculations were based on the sequence of tokens extracted from the reference text of the corresponding MR.

7.2.1 Token Cutoff. For this experiment, utterances are generated by adjusting the slot-value cutoff. As described in Section 3.1 a cutoff value was used when delexicalising the data set to ensure a minimum number of slot-types in each reference text used. For this experiment, the cutoff values used were one, two and three. A cutoff value of zero cannot be used since there would be cases when the linguistic realiser would have to try and learn a mapping from no tokens to some reference sentence.

7.2.2 Stochastic Sentence Planner. For the sentence planner, the greedy and stochastic sentence planners are compared. For this experiment, utterances were generated with a sentence planner that only uses the greedy selection of tokens and compared with those generated using stochastic selection as described in Section 4.3. Both the sentence plans produced and the utterances that were realised from the sentence plans were evaluated.

7.3 Linguistic Realiser

7.3.1 Average Checkpoints. A set of comparisons were made between utterances produced by a single model for the linguistic realiser and a model formed as an average of the previous five checkpoints for a model. These utterances were evaluated using automatic evaluation metrics.

7.4 Comparison with Template-based System

One of the problems with using automatic evaluation metrics is that there is only a weak correlation with human judgement of a text [10]. Thus human judges were used to compare the utterances generated by the data-driven and template-based systems.

⁴https://www.nltk.org/_modules/nltk/translate/bleu_score.html using smoothing function method 2

⁵<https://pypi.org/project/py-rouge>

⁶<https://pypi.org/project/jiwer/>

⁷https://www.nltk.org/_modules/nltk/metrics/scores.html

This evaluation was carried out using an online survey where each question presented the participant with two text utterances for the same MR. These texts were either generated by a data-driven system, generated by a template-based system or were the reference text for the MR. The reference texts were included as a baseline since they were written by humans and so should appear more natural. The utterances are arranged such that for each question each utterance is generated by a different method. The survey contains eight utterances generated by the data-driven and template-based systems and four reference texts. These eight utterances were the first eight utterances that the template-based system could generate from entries in the test data set. The texts were arranged as follows: six questions comparing the texts generated by the data-driven and template-based systems, two questions comparing the text generated by a data-driven system and the reference text and two questions comparing the text generated by the template-based system. These questions were arranged randomly and the participants were not given any information about how any of the pieces of text were generated. The survey was given to students aiming to collect at least 30 responses.

For each question, participants were asked to rate the clarity (how understandable and clear is the utterance) and fluency (how easy the utterance is to read) of the two texts then choose which they felt was more natural (similar appearance to human written utterances). Ratings of clarity and fluency were on a discrete scale from one to five with one being very clear/easy to read and 5 being very unclear/hard to read.

The utterances produced by the data-driven system were generated using the “greedy” sentence planner to generate the sentence plans with a slot-value cutoff of two. An average of the last five checkpoints was used for the model. These parameters for the data-driven system were chosen based on the results in Sections 8.1 and 8.2.

8.1.1 Comparison with Automatic Evaluation Metrics. To determine if automatic metrics could have been used rather than human judgement they were used to compare the utterances generated by the two systems. The BLEU, ROUGE and WER scores were calculated for the utterances generated from the first 1000 entries in the validation data set.

8 RESULTS

8.1 Sentence Planners

8.1.1 Token Cutoff. The graph in fig 3 shows the average scores for some of the metrics used to evaluate the experiment. The F1 scores are shown for the ROUGE-W and ROUGE-L metrics. The F1 scores were used since they provide a balance between the recall and precision scores. Interestingly, the BLEU score increases as the cutoff increases, while the scores for ROUGE-W, ROUGE-L, WER and missing slot-types are all best for a cutoff of one. Though the metrics, in general, do improve when increasing the cutoff from two to three, except for the BLEU metric they are all still worse than the scores achieved using a cutoff of one.

8.1.2 Stochastic Sentence Plan. To determine the quality of the sentence plans the recall, precision and F1 measures for the sentence plans were calculated for the first 1000 entries in the validation data

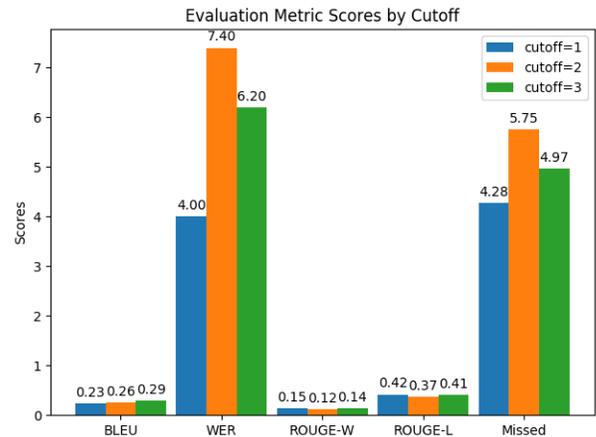


Figure 3: Evaluation metrics for utterances generated, adjusting the minimum number of tokens in the reference text

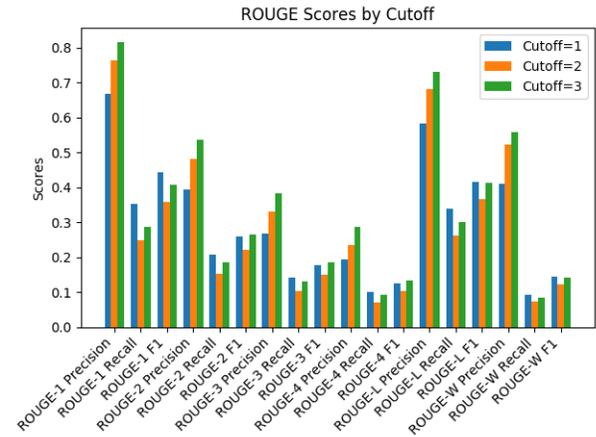


Figure 4: ROUGE scores adjusting the minimum number of tokens in the reference text

set (Table 3). Based on these results the greedy sentence planner appears to produce better sentence plans. As expected the better sentence plans produce better natural language utterances. This is based on the evaluation of the utterances, using automated evaluation metrics, generated using the different sentence planners. A summary of the results is shown in Figure 5. Following a similar pattern to the previous experiments, the BLEU score is the only metric that improves with this change. While the ROUGE and WER scores decrease and significantly more facts are omitted in the generated utterance.

8.2 Linguistic Realiser

8.2.1 Average of Checkpoints. One technique to improve the quality of the encoder-decoder used in the linguistic realiser is to take an average of several checkpoints used for the model rather than just the final model [4]. With this in mind, models were generated

	Greedy	Random
Recall(%)	99.9	93.8
Precision(%)	99.8	99.6
Accuracy(%)	84.1	88.6
F1 Measure(%)	99.8	96.4

Table 3: Sentence plan evaluation

by averaging the last five models for the models with a cutoff of one, two and three. A summary of the results is shown in Table 4. These indicate that averaging the checkpoints only improved the quality of results for the model using the cutoff of two while the performance for the models with cutoff one and three decreased based on every metric.

BLEU			WER		
Cutoff	Single	Average	Cutoff	Single	Average
1	0.227	0.258	1	3.999	7.281
2	0.257	0.287	2	7.402	6.183
3	0.289	0.287	3	6.196	6.214

Missed			ROUGE-W F1		
Cutoff	Single	Average	Cutoff	Single	Average
1	4.282	5.704	1	0.145	0.122
2	5.755	4.938	2	0.122	0.141
3	4.966	4.957	3	0.141	0.141

Table 4: Average Model Checkpoints Comparison

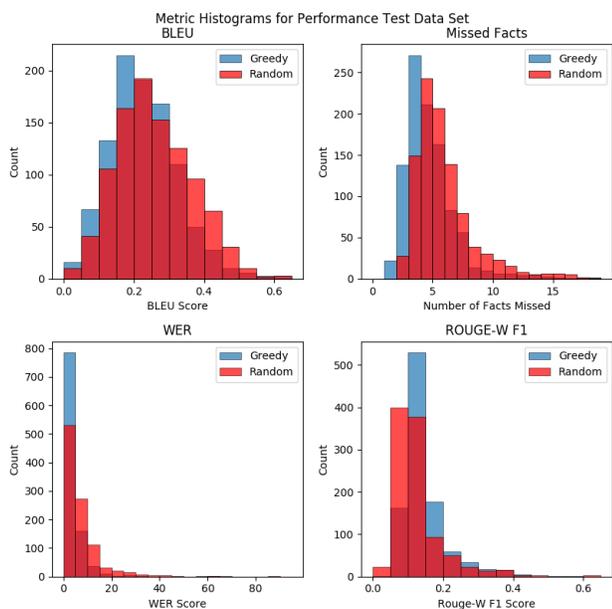


Figure 5: Comparison of utterances generated with and without random selection

8.3 Comparison with Template-based System

The survey was completed by 91 students and young adults mostly from the University of Cape Town. Figure 6 shows the mean clarity and fluency rankings for each question. The figures show that

the template-based system was rated as having higher clarity and fluency than the data-driven system for every question. Based on the survey, the template-based system seems to produce text of a similar quality to the reference-texts. For question two the template-based utterance has a higher clarity and fluency ranking than the reference text while for question four it has slightly lower rankings. The clarity and fluency of texts were rated to try and determine why

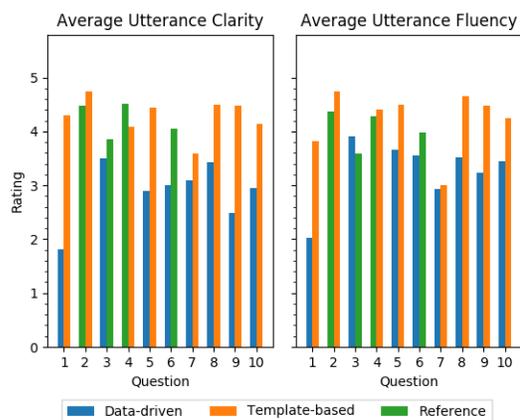


Figure 6: Clarity and Fluency ratings by question

one text was chosen as being more natural than another. Figure 7 indicates that in general the higher the clarity and fluency rating the more natural the text. Though there are a few exceptions to the rule. Furthermore, the clarity of the text appears to be a better indicator of the naturalness than the fluency based on the figure. To confirm this the Pearson’s correlation coefficient was calculated for the number of votes the data-driven and template-based systems received (appendix C). This confirmed that the clarity rating was a better indicator, particularly for the data-driven utterances. This indicates that the semantic and syntactic errors in the texts produced by the data-driven system caused participants to feel that the text was less natural. The clarity and fluency do not fully explain why one text is more natural than another since for all four questions the reference text was chosen as the more natural text even though the text was not necessarily rated as the clearest or fluent.

8.3.1 Comparison using Automatic Evaluation Metrics. The results of evaluating the utterances produced by the template-based and data-driven systems are shown in Table 5. The data-driven system achieved a significantly higher average BLEU score compared to the template-based utterances. Although the template-based system’s utterances had a lower average WER score than those produced by the data-driven system. While the Rouge scores for the two systems were similar with the data-driven system generally scoring slightly higher.

9 DISCUSSION

9.1 Sentence Planners

9.1.1 Token Cutoff. As shown in figure 3 the BLEU scores are the only metric that improves as the cutoff increases. This may be

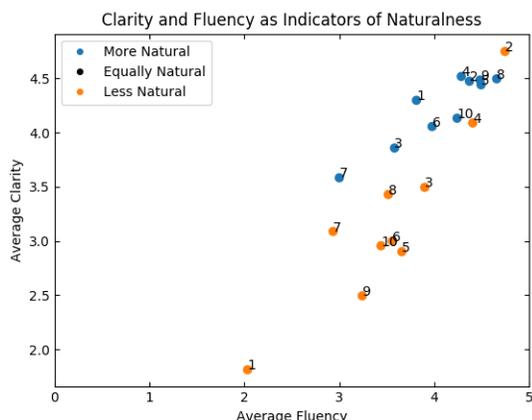


Figure 7: Average clarity and fluency for each text in each question coloured based on whether it was selected as being the more natural text and annotated with the question number

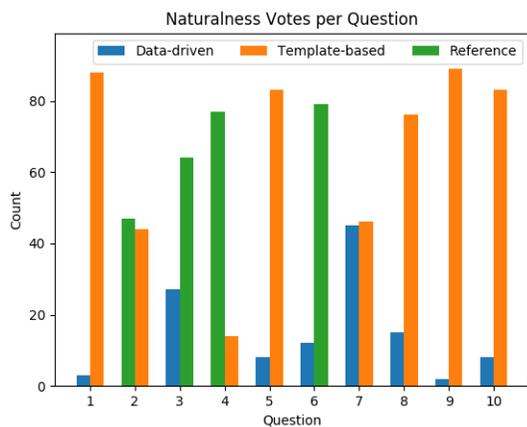


Figure 8: Number of votes each text received for being the more natural text by question

caused by some tokens appearing mostly on their own, thus the linguistic realiser struggles to learn how to incorporate them into natural language sentences so they are left out. A possible reason for the increase in BLEU score is that the realiser generates longer sentences and the BLEU score takes the length of an utterance into account.

9.1.2 *Stochastic Sentence Plan.* When using the stochastic sentence plan rather than the greedy sentence plan the BLEU scores were the only metric that increased. While the ROUGE-W and WER scores decreased and the number of missing facts increased. The decrease in performance is most likely due to the linguistic realiser struggling to realise sub-optimal sentence plans. Since the greedy sentence planner is more likely to generate sentence plans that are similar

Metric	Date-driven System	Template-based System
BLEU	0.287	0.059
WER	6.18	0.84
ROUGE-W F1	0.141	0.119
ROUGE-L F1	0.415	0.358
ROUGE-4 F1	0.134	0.104
ROUGE-3 F1	0.185	0.150
ROUGE-2 F1	0.265	0.219
ROUGE-1 F1	0.410	0.334

Table 5: Average automatic evaluation metric scores for utterances generated from the first 1000 entries in the validation data set

to the sequences of tokens used for training the encoder-decoder model in the linguistic realiser.

9.2 Linguistic Realiser

9.2.1 *Average of Checkpoints.* The results showed that averaging the last five checkpoints led to an improvement in the model with a slot-value cutoff of two while, in general, the performance of the models with a slot-value cutoff of one and three decreased. A possible reason averaging checkpoints did not yield a significant performance increase is that the number of epochs was not high enough for the models to properly converge. This suggests that the model produced with a cutoff of two converged faster than the models with cutoffs of one or three thus the quality of each of the checkpoints averaged was higher.

9.3 Comparison with Template-based System

The text for question two had higher clarity ratings than the reference text and the texts from questions two, three and four had higher fluency ratings than the reference text. This is understandable since the reference texts are scraped from Wikipedia and did not necessarily appear consecutively in the article since only the sentences containing slot-types from the MR associated with the entry are included. This lack of clarity and fluency in the reference texts is likely to make it harder for the linguistic realiser to produce good utterances since it is essentially trying to learn how to map token sequences to reference texts. Thus flawed reference texts will lead to poorer results from the realiser.

A two-sided student-t test was used to compare the mean clarity, fluency and number of votes as most natural for the utterances produced by the data-driven and template-based systems. This shows that the texts produced by the template-based system had a higher average clarity rating ($t = 6.09, p = 0.00005$), higher average fluency rating ($t = 3.273, P : 0.00556$) and on average received more votes as more for being the more natural text ($t = 4.592, p = 0.00085$).

9.3.1 *Comparison using Automatic Evaluation Metrics.* The results from the automatic evaluation metrics confirm that the use of human judges was necessary, since based on the automatic evaluation metrics the two-systems would appear to have produced texts of

a similar quality. However, the human judges found the template-based systems to have significantly higher clarity, fluency and naturalness. A possible reason for the similar metric scores is that the data-driven learns how to realise the utterances from the reference texts while the templates were handcrafted. Thus the data-driven systems utterances may have more sequences of words in common with the reference text than the template-based system.

9.4 Automatic Evaluation Metrics

Automatic evaluation metrics such as BLEU, ROUGE and WER were used to evaluate the performance of the various configurations of the NLG system. However based on the results from the survey, shown in Table 9, there is no significant correlation between the automatic evaluation metrics and the survey participants evaluation of the utterance clarity and fluency of the text. The highest correlation is between the number of facts missed as a percentage of the total number of facts in the input MR and the text clarity (-0.5) which indicates that the NLG system was better at rendering texts when there were fewer facts rendered in the text.

9.4.1 Template-based System Coverage. Though it doesn't appear in the results, the data-driven system was in general able to generalise to unseen examples better than the template-based system; since for the first 1000 entries of the validation data set the template-based system was able to produce utterances for 76.6% of the entries. The reason for this is that the template-based system was unable to generate utterances for entries that didn't have a corresponding major template.

10 CONCLUSION

It is worth noting that both systems were developed with relatively little prior knowledge over a short time span. The data-driven system was able to generalise better than the template-based system. However, the template-based system outperformed the data-driven system in terms of clarity, fluency and naturalness. This indicates that the template-based approach produced higher quality utterances, in terms of clarity, fluency and naturalness, than the data-driven system. Thus it may be better to take a template-based approach as opposed to the data-driven approach when constructing a data-to-text NLG system in order to generate higher quality utterances.

ACKNOWLEDGMENTS

Zola Mahlaza, my supervisor for his advise and guidance.
Maria Keet, for her advise on evaluating the texts.

REFERENCES

- [1] Ondřej Dušek and Filip Jurčiček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491* (2016).
- [2] Albert Gatt and Emiel Kraahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018), 65–170.
- [3] Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, 90–93.
- [4] Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. *CoRR abs/1610.01108* (2016). <http://arxiv.org/abs/1610.01108>
- [5] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, Vancouver, Canada, 67–72. <https://www.aclweb.org/anthology/P17-4012>
- [6] 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*. 1101–1112.
- [7] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Association for Computational Linguistics, Barcelona, Spain, 74–81. <https://www.aclweb.org/anthology/W04-1013>
- [8] Susan W. McRoy, Songsak Channarukul, and Syed S. Ali. 2003. An augmented template-based approach to text realization. *Natural Language Engineering* 9, 4 (2003), 381–420. <https://doi.org/10.1017/S1351324903003188>
- [9] Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation. *CoRR abs/1904.03396* (2019).
- [10] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why We Need New Evaluation Metrics for NLG. *CoRR abs/1707.06875* (2017). <http://arxiv.org/abs/1707.06875>
- [11] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254* (2017).
- [12] Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*. 463–471.
- [13] Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3, 1 (1997), 57–87.
- [14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3104–3112. <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- [15] Chris van der Lee, Emiel Kraahmer, and Sander Wubben. 2017. PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*. 95–104.
- [16] Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a Knowledge Base. In *Proceedings of the 11th International Conference on Natural Language Generation*. 10–21.
- [17] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-hao Su, David Vandyke, and Steve J. Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. *CoRR abs/1508.01745* (2015). <http://arxiv.org/abs/1508.01745>
- [18] Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755* (2015).

A DATASET STRUCTURE

Token Type	Token Value	
Name	Silvi Jan	
Date of Birth	27 October 1973	
Member of a Sports Team	ASA Tel Aviv University	
	Hapoel Tel Aviv F.C.(women)	
	Maccabi Holon F.C. (women)	
	Israel women’s national football team	
	Matches	22
	Goals	29
Country of Citizenship	Israel	
Position	Forward (association football)	

Table 6: Token Value Pairs for Silvi Jan [16]

Reference Text
Silvi Jan (born 27 October 1973) is a retired female Israeli . Silvi Jan has been a Forward (association football) for the Israel women’s national football team for many years appearing in 22 matches and scoring 29 goals. After Hapoel Tel Aviv F.C.(women) folded, Jan signed with Maccabi Holon F.C. (women) where she played until her retirement in 2007. In January 2009, Jan returned to league action and joined ASA Tel Aviv University . In 1999, with the establishment of the Israeli Women’s League, Jan returned to Israel and signed with Hapoel Tel Aviv F.C.(women)

Table 7: Reference Text for Silvi Jan [16]

Delexicalised Reference Text
Name (Date of Birth) is a retired female Israeli . Name has been a Position for the Member of a Sports Team for many years appearing in Member of a Sports Team Matches matches and scoring Member of a Sports Team Goals goals. After Member of a Sports Team folded, Jan signed with Member of a Sports Team where she played until her retirement in 2007. In January 2009, Jan returned to league action and joined Memeber of a Sports Team . In 1999, with the establishment of the Israeli Women’s League, Jan returned to Country of Citizenship and signed with Member of a Sports Team

Table 8: Delexicalised Reference Text for Silvi jan[16]

B EVALUATION METRICS

- Precision: The fraction of values from the set of values in generated text that are also in the set of values from the reference text.
- Recall: The fraction of values from the set of values in reference text that are also in the set of values from the generated text.
- F1 Measure: a combination of the recall and precision.
- Accuracy: The fraction of values from the generated text that are the same as the value in the same position in the reference text.

B.1 ROUGE

- ROUGE-N: These metrics measure the word overlaps for N-grams of a particular size. For example ROUGE-2 looks at the word overlap for pairs of words occurring in the reference and generated texts [7]. For this experiment measures are taken with N = 1,2,3 and 4.
- ROUGE-L: Focuses on the longest common sequences of words. Only in-sequence co-occurrences of words count [7].
- ROUGE-W: ROUGE using weighted longest common sequences to score the text. This builds on ROUGE-L by also taking into account the positions in which the values in the longest common sequence occur [7].

C CORRELATION COEFFICIENTS

Metric	Clarity	Fluency
BLEU	-0.302	-0.020
ROUGE-L F1	-0.142	0.019
ROUGE-W F1	-0.141	0.056
WER	-0.204	-0.131
Missing Facts(%)	-0.500	-0.044
Added Facts(%)	-0.006	0.002

Table 9: Correlation between the utterances metric score and the clarity and fluency ratings using Pearson’s correlation coefficient

Approach	Clarity	Fluency
Data-driven	0.558	0.135
Template-based	0.310	0.088

Table 10: Correlation between the number of votes as the most natural and the clarity and fluency ratings