UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2019

Title: Comparing the Utterances Generated by Template-based and Data-driven NLG Systems

Author: Matthew Poulter

Project Abbreviation: E2ET

Supervisor(s): Zola Mahlaza and Maria Keet

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	15
System Development and Implementation	*0*	*20*	15
Results, Findings and Conclusion	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	**80**

Comparing the Utterances Generated by Template-based and Data-driven NLG Systems

Matthew Poulter
University of Cape Town
pltmat001@myuct.ac.za

ABSTRACT

There are several approaches to building National Language Generation (NLG) systems. Two such approaches are a more traditional template-based approach and a data-driven approach. Each has its own strengths, and likewise, its own shortcomings, but thus far it is not certain which approach produces more natural utterances. Naturalness in this context refers to how closely text resembles natural human writing styles and nuances. The problem exists in evaluating this naturalness in order to compare these two NLG approaches successfully. This paper presents two indicators, clarity and fluency, as ways of determining the naturalness of text. In addition to this, this paper describes two systems, a data-driven system and a template-based system, which were created to generate utterances for a given Meaning Representation (MR). The utterances produced by these systems were compared using automated metrics as well as human evaluation. The results of these evaluations show that there is a significant correlation between the clarity and fluency indicators and perceived naturalness, and that there is a strong preference for template-based text over data-driven text when comparing their naturalness.

KEYWORDS

Natural Language Generation, Data-to-text Generation, Template-based Natural Language Generation, Data-driven Natural Language Generation, Text Analysis

1 INTRODUCTION

Data-to-text National Language Generation (NLG) is the process of creating natural language utterances from non-linguistic representations of information [15]. More plainly, NLG is the task of generating human-readable text from non-linguistic [15] and structured data [14].

In this way, NLG systems are used to make data easier for humans to interpret. They are already used in areas such as journalism [8], weather forecast generation [2], sports commentary [18], and medical reports [6]. These systems are all able to process large volumes of data, extracting only the relevant information, to produce human-readable text, often in a much shorter time than would be possible than a person could [4]. This obviously makes NLG systems very desirable, be it to gain a competitive advantage or to decrease the time taken for critical events.

While there are many different approaches to NLG, this investigation focuses on two of them, namely a more traditional template-based approach and the data-driven approach. Each has its own strengths, and likewise, its own shortcomings, but thus far it is not certain which approach produces more natural utterances. Naturalness in this context refers to how closely text resembles natural human writing styles and nuances. The problem, however, exists in evaluating this naturalness in order to compare these two NLG approaches successfully.

Thus, the aims of this project were to find suitable evaluation indicators for naturalness and to determine the difference in naturalness of utterances produced by data-driven and template-based NLG systems.

In terms of indicators of naturalness, this paper presents clarity and fluency of utterances as a way of determining whether one text is to be perceived as being more natural than another. In this context, clarity refers to how understandable and clear the utterance is [18] and fluency refers to how easy the utterance is to read [18].

The focal research question of this project was therefore: Does the perceived naturalness of utterances correlate with their clarity and fluency, and, further, does a data-driven approach, rather than a template-based approach, produce more natural utterances?

In order to properly compare these approaches, two systems were designed and developed - the one being a template-based system developed by the writer, and the other being a data-driven system developed by Mr Jarryd Dunn - to generate utterances for a given Meaning Representation (MR). The utterances produced by these systems were compared using automated metrics as well as human evaluation in the form of a survey.

The results of these evaluations show that there is a significant correlation between perceived naturalness and the clarity and fluency indicators. The results further show that there is a strong preference for template-based text over data-driven text when comparing their naturalness.

This paper first takes a look at the differences between these two approaches as well as the related works. Section 3 and Section 4 present the design, development, and implementation of the two systems previously mentioned. Section 5 describes how the utterances produced by the two systems were evaluated, and Section 6 looks at the results of these evaluations. Finally, the findings are presented in Section 7.

2 BACKGROUND

As has been said, the focus of this project was on comparing a traditional template-based approach with a data-driven approach to National Language Generation. This section aims to give an overview of the current known differences between these approaches, as well as discussing related works.

2.1 Definition and Comparison

Template-based text generation makes use of one or more templates, either hand-crafted or learnt from training data [14]. There are many levels of complexity available in this approach, however, one of the defining features is that data from the MRs is inserted into one or more templates within the process of producing an utterance.

On the other hand, data-driven NLG systems use machine learning techniques to learn how to produce natural language utterances from a given input. These utterances appear to be more similar to human-written text than those produced using template-based approaches [12, 22]. Additionally, data-driven systems can produce more varied outputs than template-based systems [12, 22].

Template-based systems do not tend to scale as well as data-driven approaches on large open domain systems [10, 21] since it is more difficult to generate a template that is widely applicable. This is due to the fact that template-based approaches tend to require redundant information (repeating patterns) to create the templates [20].

When compared to a template-based approach, using a data-driven approach can potentially decrease the amount of time it takes to develop an NLG system [4]. The opportunity cost of this development time improvement is that there is less control over the utterances produced by data-driven NLG systems, and 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 [4], despite these outputs appearing more similar to human-written text. Some of these semantic errors may, however, be removed by applying additional rules after generating the utterance [14].

2.2 Related Works

Puzikov and Gurevych [14] have previously compared neural models and template-based approaches for National Language Generation. They constructed a data-driven system based on TGEN [3], which they named D-model, as well as a template-based model, which they named T-model. When the utterances produced by the two systems were evaluated it was found that T-model produced no grammatical errors while there were a few punctuation errors or incorrect verbalisations in the utterances produced by D-model. D-model scored higher in all the evaluation metrics used for the comparison against T-model. That notwithstanding Puzikov and Gurevych concluded that the costs of generating complex data-driven systems may not be justified, and that problems such as theirs may be better solved using simpler techniques, referring to the template-based approach.

This project is different in that the data-driven approach was not confined to using neural techniques. This project also aimed to evaluate which utterances, those produced by a data-driven system or those produced by a template-based system, are considered more natural, as well as whether clarity and fluency were good indicators of this naturalness. Finally, this project used a more general and larger dataset than that which was used by Puzikov and Gurevych [14].

3 DATASET AND SYSTEM DESIGN

For obvious reasons, a dataset is required for the construction and evaluation of both the template-based and the data-driven NLG systems. This section describes the dataset used, as well as describing the design of each of the two systems.

3.1 Dataset

For this project the Wikipedia people dataset[1] [20] was used. This dataset is made up of a set of entries, where each entry consists of a set of slot types and their values as well as any sentences from the corresponding Wikipedia page that include at least one of the values. These sentences were used as the reference text for the entry. The slots and their values are determined from the Wikipedia information boxes. Tables 1 and 2 show an example MR.

Slot type	Slot 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 1: An example set of slot-value pairs from the Wikipedia people dataset [20]

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 2: An example reference text from the Wikipedia people dataset [20]

This dataset was both large and varied enough to allow for models and templates to be generated that were able to generalise well to unseen inputs.

The dataset was split into three sub-datasets: training (60%), validating (30%) and testing (10%). These sub-datasets were the same for both NLG systems. The training dataset was be used to learn the models and create the templates for generating utterances. These models and templates were then validated using MRs from the validation dataset. Finally once the models and templates were performing suitably well on the validation datasets they were be

[1]The Wikipedia people dataset can be found here: https://drive.google.com/open?id=1TzcNdjZ0EsLh_rC1pBC7dU70jINcsVJd

compared using MRs from the testing dataset. This was done to ensure that neither of systems had seen the input MR before, allowing a fairer comparison [7].

3.2 Data-driven Design

The data-driven system was designed by Mr Jarryd Dunn and was based on the system designed by Moryossef et al. [11]. This makes use of two modules: a sentence planner and linguistic realiser. The sentence planner makes use of non-neural techniques to try and construct a symbolic text plan based on the input MR. The linguistic realiser uses neural generation to turn the text plan into a text utterance.

For both modules, either a reinforcement learning or neural approach may be used to learn a model. The data-driven system was designed in a similar way to Moryossef et al. [11], with the first module making use of a non-neural approach to generate the text plan. A second module uses a neural approach to realise the text into a linguistic utterance. The output of the first module consists of a sequence of tokens that determine the underlying structure of the utterance (what information it will include and how this information will be ordered). The second module then expresses the content from the text plan in natural language. This design was chosen since systems built using non-neural techniques (such as reinforcement learning) tend to produce fewer errors but also less variation [7]. This suggests that constructing one module using non-neural techniques and another using neural techniques may produce a system that generates varied utterances with few errors [11].

In order to simplify the construction of the system, a prebuilt encoder-decoder module called OpenNMT[2] was used to construct the neural networks. This allowed more focus to be put on the actual text planning and realisation.

3.3 Template-based Design

Given the nature of this system, templates needed to be created for use in generating the utterances. These templates were hand-crafted using the training sub-dataset. The creation process followed a similar approach to van der Lee et al. [18], such that entries within the dataset were categorised according to the data available in the slots and the way this data is conveyed in the reference text.

To assist with the template creation, a retrieval-based method of finding similarities between the reference texts was used, as described by van der Lee et al. [19]. This method uses cosine similarity to assign a score representing how similar two input MR were. Similar input MRs were then clustered together, requiring fewer templates to be created.

From the categorisation of the templates, major and minor template categories were formed such that minor templates were grouped together into a major template according to which slots were required for the template. The major templates were therefore categorised by these required slots. An example of a major template with corresponding minor templates is shown in Table 3. This dual-layer categorisation aimed to allow for two things:

(1) the system can lookup a set of templates by which slots are available in the input MR; and

(2) redundant templates can be added to increase variety in the utterances provided.

Slots	Name ; Date of Birth ; Member of a Sports Team ; Country of Citizenship ; Position
Minor template	< Name >(born < Date of Birth >) is from < Country of Citizenship >. <o:Sex or Gender:He:She>was a < Position >for many teams, including <l: Member of a Sports Team :and>.
Minor template	< Name >(born < Date of Birth >) was a < Position >for <l: Member of a Sports Team :and>. <o:Sex or Gender:He:She> is from < Country of Citizenship >

Table 3: An example major template used in the template-based system.

Finally, the templates were stored in JSON structures where each sentence was broken into its different parts of speech (such as the subject, verb and object). This mirrors the inputs required for the library which was used in the realising of the final utterance and is discussed more below.

The architecture of the template-based NLG system loosely follows the tri-module pipeline approach as proposed by Reiter and Dale [16]. Figure 1 has been included here as a reference to this, and is to be considered in the context of the design of the system.

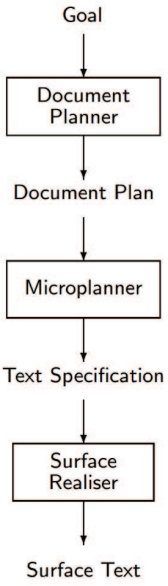

Figure 1: NLG system architecture as proposed by Reiter and Dale [16]

The Document Planner module evaluates the input MR provided by creating a list of the slot types available in the MR. It then

[2]http://opennmt.net/

selects an appropriate major template from the template storage by finding the major template with the most number of matching slot types. Using this template and the input MR, the module produces a Document Plan custom data structure.

This Document Plan is passed to the Microplanner module which randomly chooses one of the redundant list of minor templates. From this, a Sentence Plan custom data structure is produced containing a full specification of the text. This Sentence Plan includes a list of sentences, broken up into clauses, and further split into the subject, object, verb and tense of the sentence, along with any sub-clauses and complements. This parallels the inputs required for the library used in the Realiser module.

The Realiser module takes the Sentence Plan and uses the SimpleNLG API library [5] to produce the utterance. The SimpleNLG library is capable of creating well-formed, grammatically and syntactically correct sentences given a set of inputted components. As previously mentioned, the templates themselves, as well as the Sentence Plan, are in formats which parallels the inputs required for the SimpleNLG library.

This template-based system design was created with the Java development language in mind, and the system itself was developed in Java, as is discussed in the following section.

4 SYSTEM DEVELOPMENT

This paper does not go into too many details about the data-driven system developed by Mr Jarryd Dunn, save to say that it was developed in Python using the OpenNMT module. To find out more about his system, it is suggested that his paper on the same topic be read.

In terms of the development of the template-based system, this was developed in Java by the writer, and the implementation is discussed in detail below.

4.1 Implementation

To facilitate running the three modules using the pipeline design pattern, as described in Section 3, an implementation of the pattern example by Bapat [1] was used. This allowed for the three modules to be easily added as shown in the example code in Figure 2. This code was put in the main method and class of the system. The input to the pipeline is a single line of the dataset file as each line of the dataset represents a different MR.

```
Pipeline<String, String> pipeline =
    new Pipeline<>(new DocumentPlanner())
            .pipe(new MicroPlanner())
            .pipe(new SurfaceRealiser());

String output = pipeline.execute(input);
```

Figure 2: Adding three modules to the Java pipeline implementation by Bapat [1].

In addition to running the pipeline, the main method is also responsible for reading in all the major templates from previously stored text files containing representations of templates such as the example shown in Table 3. Each text file represents one major template, where the first line contains a comma-separated list of slot types needed, and every subsequent line contains a single minor template in JSON format. An example JSON formatted minor template can be seen in Figure 3.

```
[
  {
    "clauses": [
      {
        "subject": "<Name> (born <Date of
                    Birth>)",
        "verb": "is",
        "tense": "PRESENT",
        "object": "from <Country of
                   Citizenship>",
        "subclauses": [],
        "complements": []
      }
    ]
  },
  {
    "clauses": [
      {
        "subject": "<o:Sex or Gender:He:
                    She>",
        "verb": "is",
        "tense": "PAST",
        "object": "a <Position> for many
                   teams, including
                   <l:Member of a Sports
                   Team:and>",
        "subclauses": [],
        "complements": []
      }
    ]
  }
]
```

Figure 3: An example JSON formatted minor template.

Within the Document Planner module, the planner uses a JSON parsing library, JSON Simple [3], to read the line of the input dataset file into an object. The available slot types are then read and put into a list. From here, the system iterates through all the major templates available, counting the number of slot types similar to the available inputted slot types, and excluding any major templates which contain slot types not available in the input MR.

Finally, the system selects the major template which contains the most number of slot types overlapping, producing a Document Plan similar to the code in Figure 4. This includes an object mapping the

[3] https://code.google.com/archive/p/json-simple/

slot types to their relevant values from the input MR. If no major template can be found, the system throws and error.

```
class DocumentPlan {
    MajorTemplate majorTemplate;
    Map<String, String> slots;
}
```

Figure 4: The barebones code of the Document Plan produced by the Document Planner.

The Micro Planner module accepts the Document Plan as its input. The first thing it does is select one of the Micro Templates at random from the major template. Secondly, the Micro Planner parses the JSON of the Micro Template into a tree-like data structure similar to the example in Figure 5. Finally, the module substitutes in the relevant slot values in the positions supplied by the template.

```
class SentencePlan {
    MinorTemplate minorTemplate;
    List<Sentence> sentences;

    class Sentence {
        List<Clause> clauses;

        class Clause {
            String subject;
            String verb;
            Tense tense;
            String object;
            List<Clause> complements;
            List<Clause> subClauses;
            String complementiser;
        }
    }
}
```

Figure 5: The barebones code of the Sentence Plan produced by the Micro Planner.

Finally, the Surface Realiser module accepted the Sentence Plan as its input, traversing through the tree-like data structure and inputting each property into the SimpleNLG API library [5]. The sentences produced by the library were then concatenated into a paragraph which was return to by outputted.

4.2 Template Coverage

As mentioned in section 3, the templates were designed using a technique described by van der Lee et al. [19]. While this technique did automate the process of finding similarities in the reference texts, the templates themselves still needed to be hand-crafted. It is worthwhile to note that this was the most timely exercise in developing the template-based system.

Importantly, because the templates were hand-crafted, time was the limiting factor in how many could be created. Thus, templates were created until the system was able to produce a outputs for 75% of the inputted training data.

4.3 Validation

To validate that the output of the template-based system during development, several unit and feature tests were written. Each test provided a mock input to each of the modules in the pipeline and the system as a whole, and compared the output produced with what was expected. While the system itself was not overly complicated, and the output could for the most part be validated by hand, this testing approach allowed for confidence in refactoring and improving the modules.

Further to these unit and feature tests, several automated evaluation metrics were used to gauge how well the system output represented the input when compared to the reference text provided by the dataset. The evaluation metrics that were used were:

- BLEU [13], which measures the accuracy of the information represented in the utterance compared to the ground reference [7].
- ROUGE [9], a word-overlap metric like BLEU but which doesn't consider the length of utterances [4].
- WER (Word Error Rate), which measures the number of insertions, deletions, transpositions and substitutions required to convert the utterance to the reference text [4].

5 HUMAN EVALUATION

Since one of the main purposes of National Language Generation is to make data easier for humans to interpret, and given the research question asks, "Does the perceived naturalness of utterances correlate with their clarity and fluency, and, further, does a data-driven approach, rather than a template-based approach, produce more natural utterances?", the final evaluation of the systems was accomplished using human judges.

5.1 Survey Design

This evaluation took the form of a survey where each question showed two different texts relating to the same input MR. The participants were first asked to rate the clarity and fluency of each of the texts, and then asked to choose which of the two texts they found more natural. They were also provided with definitions for the terms clarity, fluency and naturalness, within the context of this research.

Several questions were included which contained one utterance generated by one of the systems along with the original reference text, rather than the utterance from the other system. It was expected that the participant would judge the reference text to be more natural, thus giving an indication of the reliability of the participant's judgement.

The participants were not shown the MR used to generate the utterances, since this might cause them to focus more on the inclusion of information rather than the naturalness of the text [17].

5.2 Execution

After being approved by the University of Cape Town (UCT) Faculty of Science Research Ethics Committee and the UCT Department of Student Affairs (DSA), the survey was uploaded to Google Forms and was distributed to students of the university via the DSA research mailing list. In addition to this, the survey was distributed to various other student and younger adult groups to gain a broader sampling of responses.

Participants were able to complete the survey online after reading the appropriate university's Informed Voluntary Consent form and giving their consent for their answers to be recorded and used in this research project. The results were tabulated using a spreadsheet and are analysed below.

6 RESULTS

This section presents the results from the coverage testing, the automated metrics, and the human evaluation.

6.1 Coverage

Various coverage metrics were recorded for each of the two systems for the first 1000 MRs in the validation sub-dataset and are presented in Table 4.

	Template	Data-Driven
No. of inputs	1000	1000
Coverage	76.60%	100%
Average no. of sentences in reference text	9.111	9.111
Average no. of sentences in output text	1.393	1.914
Average no. of slot types in MR	8.688	8.688
Average no. of slot types not used in output	3.275	4.938
Average no. of fictional slot types added to output	0	0.589

Table 4: The coverage of the template-based system in finding a suitable template and generating an output.

6.2 Automated Metrics

The automated evaluation metrics were run on both systems, and Table 5 shows the results of these metrics when run on the first 1000 MRs in the validation sub-dataset.

6.3 Human evaluation

In total, 98 people completed the survey providing an appropriately sized sample of results. This subsection provides a summary of these results.

As was described in Section 5, the survey asked the participants to rate the text's clarity and fluency on a scale of 1 to 5. Table 6 shows the mean ratings for each of the texts generated by the template-based system, as well as the overall mean and standard deviation from the mean for all the generated texts.

	Tempate	Data-Driven
rouge-1-p	0.85759	0.81406
rouge-1-r	0.22324	0.28889
rouge-1-f	0.33409	0.40971
rouge-2-p	0.57701	0.53481
rouge-2-r	0.14647	0.18599
rouge-2-f	0.21902	0.26461
rouge-3-p	0.41023	0.38021
rouge-3-r	0.10006	0.12955
rouge-3-f	0.14990	0.18488
rouge-4-p	0.29951	0.28464
rouge-4-r	0.06911	0.09353
rouge-4-f	0.10404	0.13437
rouge-l-p	0.78399	0.72946
rouge-l-r	0.24633	0.30208
rouge-l-f	0.35843	0.41480
rouge-w-p	0.63420	0.55761
rouge-w-r	0.06870	0.08394
rouge-w-f	0.11857	0.14119
wer	0.84013	6.18257
bleu	0.05945	0.28719

Table 5: Results of automated metrics on first 1000 MRs of the validation sub-dataset.

Question	Clarity	Fluency
1	4.30	3.79
2	4.74	4.73
4	4.09	4.39
5	4.45	4.51
7	3.55	2.97
8	4.48	4.65
9	4.49	4.48
10	4.14	4.23
Mean	**4.28**	**4.22**
Std Dev	**0.36**	**0.58**

Table 6: Mean clarity and fluency ratings (out of 5) for the texts generated by the template-based system.

Question	Clarity	Fluency
1	1.81	2.07
3	3.50	3.92
5	2.88	3.63
6	2.99	3.56
7	3.07	2.94
8	3.39	3.50
9	2.48	3.20
10	2.95	3.43
Mean	**2.88**	**3.28**
Std Dev	**0.54**	**0.57**

Table 7: Mean clarity and fluency ratings (out of 5) for the texts generated by the data-driven system.

Table 7 shows the mean ratings for each of the generated texts, as well as the overall mean and standard deviation from the mean for texts generated by the data-driven system.

Question	Clarity	Fluency
2	4.48	4.38
3	3.89	3.60
4	4.52	4.29
6	4.03	3.94
Mean	**4.23**	**4.05**
Std Dev	**0.32**	**0.35**

Table 8: Mean clarity and fluency ratings (out of 5) for the reference texts.

Finally, Table 8 shows the mean ratings for each of the reference texts used in the survey, as well as the overall mean and standard deviation from the mean for these texts.

Prefer Over	Template Reference	Data-Driven Reference	Template Data-Driven
100%	10.2%	4.1%	39.8%
≥75%			80.6%
≥50%	53.1%	36.7%	99.0%
≥25%			100.0%
≥0%	100.0%	100.0%	100.0%

Table 9: The percentage of participants who preferred one text (first row) over another (second row) for a percentage of questions.

In terms of the preferences between the various texts, the results have been summarised in Table 9. The table shows the percentages of participants who preferred one text over another for a percentage of questions. As an example, the table should be read as "80.6% of participants preferred template-based text over data-driven text at least 75% of the time."

Correlation	Clarity	Fluency
Preference	0.826	0.856

Table 10: The Pearson correlation coefficient between the difference in clarity and fluency ratings and the percentage preference of naturalness between template-based and data-driven text.

Two Pearson correlation coefficients were calculated and are shown in Table 10. These are:

(1) the correlation between the difference in clarity ratings and the percentage preference between template-based and data-driven text; and
(2) the correlation between the difference in fluency ratings and the percentage preference between template-based and data-driven text.

7 FINDINGS

Several important findings have been made from these results and they are discussed here.

7.1 The data-driven system performed better than the template-based system in a majority of the automated metrics.

The metric results supported the observation by Puzikov and Gurevych [14] that if the templates are similar to the reference texts, the metrics will return very high results, but if they are not, they will return very low results when compared to data-driven utterances.

7.2 The template-based system produced text which was rated more clear and fluent than the data-driven system.

It is quite clear from the results that the texts from the template-based system were rated higher than those from the data-driven system in terms of clarity and fluency. What is interesting to note, however, is that the data-driven texts were felt to be significantly more fluent than clear. In contrast, the template-based texts were felt to be marginally more clear than fluent.

Further, while there was little deviation from the mean in terms of clarity in the template-based texts, the deviation of fluency ratings was the highest of all six standard deviations measured in Section 6. When evaluating why this was the case, it was observed that the template-based text in question 7 of the survey included the sentence:

> "He played for the A.F.C. Bournemouth, Cheltenham Town F.C., Molesey F.C., Croydon F.C., Clapton F.C., Croydon F.C., Hayes F.C., Yeading F.C., Fisher Athletic F.C., Grays Athletic F.C., Eastbourne Borough F.C., Dover Athletic F.C. and Cray Wanderers F.C."

This is obviously a very long and unnecessary list which the template-based system had not been able to present in a more natural way, given the input MR. This can be seen as a limitation of the system.

7.3 The template-based system produced text which was rated on par with the reference text in terms of clarity and fluency.

Interestingly, the mean ratings for clarity and fluency were similar for the template-based texts and the reference texts. When taking the mean of these ratings for the two questions directly comparing these two text types, questions 2 and 4, clarity was rated at 4.42 and 4.50 for template-based and reference text respectively. Fluency was rated at 4.56 and 4.33 for template-based and reference text respectively. This indicates that template-based systems are certainly capable of producing text which is on par with human generated text.

Of note, however, is that of the two questions, questions 2 and 4, 52.02% and 84.69% of participants (respectively) found the reference text to be more natural. This shows that clarity and fluency cannot be considered the only two factors which influence the naturalness of text.

7.4 There is a strong correlating between clarity and fluency, and the feeling of naturalness of a given text.

When comparing the differences in clarity ratings and the differences in fluency ratings with the preference participants had when selecting which text felt more natural, one can see a strong positive correlation, with a Pearson coefficient of 0.83 for clarity and 0.86 for fluency. This shows that these are good indicators to be used when evaluating NLG systems in terms of naturalness.

7.5 There is a strong preference for template-based text over data-driven text when evaluating for naturalness.

Of the six questions comparing template-based text with data-driven text, five of the template-based texts were felt to be more natural by at least 80% of the participants. The sixth question, however, was question 7 of the survey and saw an exact 50% division in the participants. This is the same question discussed in Section 7.2. This text from the template-based system can therefore be considered an outlier.

When excluding this question, the mean percentage of participants who found the template-based texts to be more natural is 92.4% with a standard deviation of 5.7%. This clearly shows a strong preference for template-based text over data-driven text when evaluating for naturalness.

8 CONCLUSIONS

The focal research question of this project was: Does the perceived naturalness of utterances correlate with their clarity and fluency, and, further, does a data-driven approach, rather than a template-based approach, produce more natural utterances?

Having analysed the results of the survey comparing outputs of the two designed systems, this paper has shown that there is a significant correlation between perceived naturalness and the clarity and fluency indicators. The results further show that there is a strong preference for template-based text over data-driven text when comparing their naturalness.

REFERENCES

[1] Deepak Bapat. 2019. The Pipeline design pattern (in Java). https://medium.com/@deepakbapat/the-pipeline-design-pattern-in-java-831d9ce2fe21

[2] Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering* 14, 4 (2008), 431–455. https://doi.org/10.1017/S1351324907004664

[3] Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491* (2016).

[4] Albert Gatt and Emiel Krahmer. 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.

[5] 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 '09)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 90–93. http://dl.acm.org/citation.cfm?id=1610195.1610208

[6] Albert Gatt, Ielka Van Der Sluis, and Kees Van Deemter. 2007. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*. Association for Computational Linguistics, 49–56.

[7] 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.

[8] Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*. 188–197.

[9] 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

[10] 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

[11] 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).

[12] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254* (2017).

[13] 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 (ACL '02)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[14] 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.

[15] Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3, 1 (1997), 57–87.

[16] Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, New York, NY, USA.

[17] Kyle Richardson, Sina Zarrieß, and Jonas Kuhn. 2017. The Code2Text Challenge: Text Generation in Source Libraries. In *Proceedings of the 10th International Conference on Natural Language Generation*. Association for Computational Linguistics, Santiago de Compostela, Spain, 115–119. https://doi.org/10.18653/v1/W17-3516

[18] Chris van der Lee, Emiel Krahmer, 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.

[19] Chris van der Lee, Emiel Krahmer, and Sander Wubben. 2018. Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods. 35–45. https://doi.org/10.18653/v1/W18-6504

[20] 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.

[21] 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). arXiv:1508.01745 http://arxiv.org/abs/1508.01745

[22] 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).