Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods

Chris van der Lee
Tilburg University
c.vdrlee@tilburguniversity.edu

Emiel Krahmer
Tilburg University
e.j.krahmer@tilburguniversity.edu

Sander Wubben
Tilburg University
s.wubben@tilburguniversity.edu

Abstract

The current study investigated novel techniques and methods for trainable approaches to data-to-text generation. Neural Machine Translation was explored for the conversion from data to text as well as the addition of extra templatization steps of the data input and text output in the conversion process. Evaluation using BLEU did not find the Neural Machine Translation technique to perform any better compared to rule-based or Statistical Machine Translation, and the templatization method seemed to perform similarly or sometimes worse compared to direct data-to-text conversion. However, the human evaluation metrics indicated that Neural Machine Translation yielded the highest quality output and that the templatization method was able to increase text quality in multiple situations.

1 Introduction

Most approaches to data-to-text generation fall into one of two broad categories: rule-based or trainable (Gatt and Krahmer, 2018). Rule-based systems are often characterised by a template-based design: texts with gaps that can be filled with information. The application of these templates generally results in high quality text (e.g. van Deemter et al., 2005). The text quality of trainable systems — e.g. statistical models that select content based on what is the most likely realization according to probability — is generally lower (Reiter, 1995) and their development slower (Sanby et al., 2016). However, trainable systems use data-driven algorithms and do not rely on manually written resources for text generation, while most template systems require manually written templates and rules for text generation. This makes trainable systems potentially more adaptable and maintainable. Different approaches have been tried to decrease the building time and cost of data-to-text systems associated with trainable approaches, while limiting the drop in output quality compared to rule-based data-to-text systems (e.g. Adeyanju, 2012; Liang et al., 2009; Mahapatra et al., 2016) by experimenting with the trainable method.

The goal of the current study was to explore the combination of template and trainable approaches by giving statistical and deep learning-based systems templatized input to create templatized output. The more homogeneous nature of this templatized form was expected to make production of output that is fluent and clear as well as an accurate representation of the data more feasible compared to their untemplatized counterpart, generally used for trainable approaches. Furthermore, the usage of statistical and deep learning methods reduces the reliance on manually written resources that is associated with most template based systems. The approach of the current study was tested on four corpora in the sports and weather domain, each with divergent characteristics, to assess the usefulness in different situations. The output of these systems is compared using automated metrics (i.e. BLEU) as well as human evaluation.

2 Background

2.1 Data-to-text

Historically, most data-to-text systems use rule-based approaches which select and fill templates in order to produce a natural language text (e.g. Goldberg et al., 1994; van der Lee et al., 2017) and these approaches are still the most widely used in practical applications (Gkatzia, 2016). This is partly because rule-based approaches are robust

Proceedings of The 11th International Natural Language Generation Conference, pages 35–45,
Tilburg, The Netherlands, November 5-8, 2018.

and can produce high quality output given sufficient development time and cost. In addition, the output of these approaches is fully controlled by humans, which make them generally accurate in their representation of the data (e.g. van der Lee et al., 2018). However, capturing data using rules may be feasible for simple situations, but reports in several domains often describe more complex situations which would require an extensive set of rules. Writing these rules is time intensive and covering all distinct rules is nearly impossible for many situations. Furthermore, developing and maintaining these systems is cost intensive and most systems are difficult to extend to other domains. Statistical approaches may provide a solution for these shortcomings. These approaches are trained using a parallel corpus, thus require no handcrafted rules. This also makes conversion to other domains less time-intensive compared to rule-based approaches.

2.2 Trainable approaches

Producing output by using such trainable approaches can be exercised in different ways. Retrieval-based models (e.g. Adeyanju, 2012), statistical approaches, such as Hidden Markov Models (e.g. Barzilay and Lee, 2004; Liang et al., 2009), and classification methods (Duboue and McKeown, 2002; Barzilay and Lapata, 2005) have all been successfully implemented. Another way of approaching the problem is by treating it as a translation challenge, where a machine translation system translates a data representation string into a target language string. Several authors have implemented Statistical Machine Translation (SMT) methods to generate natural language using aligned data-text test sets (e.g. Wong and Mooney, 2007; Belz and Kow, 2009, 2010; Langner et al., 2010; Pereira et al., 2015) all obtaining promising results. Furthermore, an SMT model was consistently among the higher scores in the WEB NLG Challenge, where the goal is to convert RDF data to text (Castro Ferreira et al., 2017; Gardent et al., 2017), thus showing the potential of SMT-based methods as a viable approach to data-to-text NLG. However, this SMT approach was less successful in other studies in which the SMT-based method was often outscored by other statistical approaches according to automated metrics as well as human evaluation (Belz and Kow, 2010).

The impressive performance of deep learning methods on various tasks such as text summarization and machine translation suggests that Neural Machine Translation methods (NMT) might have the potential to outperform its SMT counterpart. This is also supported by results in the WEB NLG Challenge where NMT approaches obtained the highest scores on automated metrics and among the highest on human evaluation. Wiseman et al. (2017) found that various Neural data-to-Text models performed relatively well on automated metrics as well as human evaluations, although they still noted a significant performance gap between these models and their baselines.

One possible reason for this performance difference Wiseman et al. (2017) found might be the nature of the datasets used. The authors noted that their data for one corpus was noisy and that many texts contained information that was not captured in the data. Other authors have also noted that the dataset is often a bottleneck of most trainable approaches, since many aligned data-text corpora are relatively small (Richardson et al., 2017). Furthermore, several data-text aligned corpora used for these tasks are the input and output of a (rule-based) data-to-text system, which means that experiments using these corpora are performing reverse-engineering and that these results may not reflect performance on human-written datasets (Reiter, 2017).

2.3 Current work

The current work investigated the potential limitations of automatically generated corpora by using several corpora with differing characteristics, but also attempted to address the issue of small datasets by exploring *templatization* as a possible solution. Templatization is similar to what others call a delexicalization step, which means that an extra step was added in the conversion from data to text: using simple rules, gaps were added in place of the data points in the aligned data and text documents. After this step, SMT and NMT techniques were trained on the aligned data-text set and new templates were produced. Finally, these templates were filled based on a similar ruleset that was responsible for templatizing the data and texts. By using such an approach, the data and texts are likely to become more homogeneous, which could help trainable approaches to find data-text connections more quickly. This means that the trainable approaches could be more robust on smaller datasets and datasets with high variety in language. Whether this hypothesis holds true is also investigated using BLEU scores as well as human assessment on clarity, fluency and correctness.

Combining trainable approaches with a template representation has been done previously, but such systems are scarce. Kondadadi et al. (2013) are one of the first and only researchers that have attempted this combination. However, their research experimented with automated sentence templatization and sentence aggregation rather than automatically generated sentences from data points. The aim of the current work can be seen as an exploratory first step in building a system that integrates these other automation techniques to generate text from data in a fully unsupervised fashion.

	Weather.gov	Prodigy-METEO	Robocup	Dutch Soccer
Lines	29,792	601	1,699	6,414
Words	258,856	6,813	9,607	116,796
Tokens	955,959	32,448	45,491	524,196
Domain	Weather	Weather	Sports	Sports
Writer type	Computer	Human	Computer	Human

Table 1: Characteristics of the (text-part of the) corpora used in this study.

3 Datasets and approaches

3.1 Datasets

A total of four different datasets were used in the current study, two datasets contain weather reports and two contain sports reports. Furthermore, one weather dataset and one sports dataset contain texts that resulted from (mainly) rule-based data-to-text generation, while the other weather and sports datasets contain human-written texts. Characteristics of these datasets are described in Table 1 and below.

3.1.1 Weather.gov

For this dataset, Liang et al. (2009) collected weather forecasts from http://www.weather.gov. These weather forecasts contain information on weather aspects, such as temperature, wind speed, and cloudiness. The original data representation was modified to reduce noise and to ensure that the data input representation and text documents both represented the same data. Furthermore, tags were added since previous research found this to be the representation resulting in the highest quality output (Belz and Kow, 2010). The complete forecast texts were reduced to the first sentence to enable equal sentence-based data-to-text generation across all domains. This resulted in a total of 29,792 data-text pairs. The texts were most likely computer-generated, with possibly some human post-processing (Reiter, 2017).

Data type	Example
Original input representation	temperature.time:17-30 temperature.min:24 temperature.mean:28 temperature.max:38 (...) sleetChance.mode:–
Tagged input representation	skyCover_mode: 0-25 temperature_minmeanmax temperature_mode: 24-28-38
Templatized tagged input representation	skyCover_mode: <cloud_data> temperature_minmeanmax temperature_mode: <temperature>
Retrieval (direct)	mostly clear , with a low around 21 .
Retrieval (templatized)	<cloud_data> , with a <high_near_low_around_steady_temperature> <temperature> .
Retrieval (filled)	sunny , with a high near 38 .
SMT (direct)	mostly clear , with a low around 22 .
SMT (templatized)	<cloud_data> , with a <high_near_low_around_steady_temperature> <temperature> .
SMT (filled)	sunny , with a high near 38 .
NMT (direct)	mostly clear , with a low around 22 .
NMT (templatized)	<cloud_data> , with a <high_near_low_around_steady_temperature> <temperature> .
NMT (filled)	sunny , with a high near 38 .

Table 2: Examples of the (original and applied) data representation and text output examples for the Weather.gov corpus

3.1.2 Prodigy-METEO

Data type	Example
Original input representation	[[1,_SSW,10,14,-,-,0600],[2,_WSW,14,18,-,-,1200], [3,_W,10,14,-,-,0000]]
Tagged input representation	WindDir.1: SSW WindSpeedMin.1: 10 WindSpeedMax.1: 14 Time.1: 0600 (...) Time.3: 0000
Templatized tagged input representation	WindDir.1: <wind_direction> WindSpeed.1: <wind_speed_min> WindSpeed.1: <wind_speed_max> (...) Time.3: <time>
Retrieval (direct)	ssw 10-14 veering wsw 14-18 by midday easing w'ly 10-14 by late evening
Retrieval (templatized)	<wind_direction> <wind_speed> <wind_direction_change> <wind_direction> <wind_speed> <time> , <wind_speed_change> <wind_direction> <wind_speed> <time>
Retrieval (filled)	ssw 10-14 veering wsw 14-18 by midday, rising w 10-14 by late evening
SMT (direct)	ssw 10-14 veering wsw 14-18 by midday easing w'ly 10-14 by late evening
SMT (templatized)	<wind_direction> <wind_speed> <wind_direction_change> <wind_direction> <wind_speed> <time> <wind_direction_change> <wind_direction> <wind_speed> <time>
SMT (filled)	ssw 10-14 veering wsw 14-18 by midday veering w 10-14 later
NMT (direct)	ssw 10-14 veering wsw 14-18 by midday easing w'ly 10-14 by late evening
NMT (templatized	<wind_direction> <wind_speed> <wind_direction_change> <wind_direction> <wind_speed> <time> then <wind_direction_change> <wind_direction> <wind_speed> <time>
NMT (filled)	ssw 10-14 veering wsw 14-18 by afternoon then veering w 10-14 later

Table 3: Examples of the (original and applied) data representation and text output examples for the Prodigy-METEO corpus

Prodigy-METEO — a dataset derived from SumTime-Meteo — was used as the second weather dataset (Belz, 2008; Sripada et al., 2002). This dataset contains human-written texts on wind data. The dataset contains a total of 601 lines. The original input vector representation was also modified to a tagged input representation inspired by the tagged input vector of Belz and Kow (2010).

Data type	Example
Original input representation	badPass.arg1: purple11 badPass.arg2: pink9 turnover.arg1: purple11 turnover.arg2: pink9
Tagged input representation	turnover.arg1: purple11 turnover.arg2: pink9 badPass
Templatized tagged	turnover.arg1: <player_1_team_1> turnover.arg2: <player_1_team_2> badPass
Retrieval (direct)	purple11 tries to pass to purple10 but was picked off by pink3
Retrieval (templatized)	<player_1_team_1> turned the ball over to <player_1_team_2>
Retrieval (filled)	purple11 makes a bad pass that picked off by pink9
SMT (direct)	purple11 makes a bad pass that was intercepted by pink9
SMT (templatized)	<player_1_team_1> makes a bad pass that was picked off by <player_1_team_2>
SMT (filled)	purple11 makes a bad pass that was picked off by pink9
NMT (direct)	purple11 loses the ball to pink9
NMT (templatized)	<player_1_team_1> makes a bad pass that was picked off by <player_1_team_2>
NMT (filled)	purple11 makes a bad pass that was picked off by pink9

Table 4: Examples of the (original and applied) data representation and text output examples for the Robocup Sportscasting corpus

3.1.3 Robocup Sportscasting

This dataset — created by Chen and Mooney (2008) — provides data and texts on the 2001-2004 Robocup finals. Each sentence represents one match event and commentary fragment of the game. These sentences were created using a data-to-text system. The original dataset was slightly altered by removing data-text lines where the data did not (fully) represent the content of the text and a tagged input representation similar to the other datasets was added, resulting in 1699 aligned data-text lines. These lines represent match events such as passes, goals, interceptions, tackles, and possession.

3.1.4 Dutch Soccer

In addition to the other corpora, an aligned data-text corpus was created for this work. Dutch soccer reports on Eredivisie matches from the 15/16 and 16/17 season were scraped from ten news websites. Additionally, texts from the Dutch part of the MeMo FC corpus (Braun et al., in press) were added, which are soccer texts scraped from Eredivisie club websites (season 15/16). Data was scraped from Squawka for these Eredivisie seasons.[1] After collecting the data, a script was written to tokenize the soccer reports, align the match data and soccer matches, and connect match sentences to specific match events in the data input using data-text overlap and simple word occurrence heuristics. After running this alignment script, a total of 6,414 sentences were found that described goals (regular, penalty and own goals), yellow cards and red cards.

Data type	Example
Original input representation	–
Tagged input representation	CardPlayer_Name: Jurgen Mattheij CardPlayer_Surname: Mattheij (...) OtherTeam: Roda JC Kerkrade ; home
Templatized tagged input representation	CardPlayer_Name: <card_player> CardPlayer_Surname: <card_player> (...) OtherTeam: <other_team>
Retrieval (direct)	rood voor jurgen mattheij , die wel knuffels van zijn ploeggenoten ontvangt *red card for jurgen mattheij, who does receive hugs from his teammates*
Retrieval (templatized)	<team> speelden <time> met tien man na een rode kaart voor <card_player> en bezitten momenteel de twaalfde positie in de eredivisie . *<team> played with ten men <time> after a red card for <card_player> and posses the twelfth place in the eredivisie .*
Retrieval (filled)	excelsior speelden vlak voor tijd met tien man na een rode kaart voor Jurgen Mattheij en bezitten momenteel de twaalfde positie in de eredivisie . *excelsior played with ten men at the end after a red card for Jurgen Mattheij and posses the twelfth place in the eredivisie .*
SMT (direct)	na een rode kaart voor een Jurgen Mattheij Mattheij ook bij de eerste rode kaart kreeg , maar zijn roda jc . *after a red card for Jurgen Mattheij Mattheij also received with the first red card , but his roda jc .*
SMT (templatized)	na <time> minuten met tien man verder nadat <card_player> met zijn eerste doelpunt voor <team> *continue with ten men after <time> minutes after <card_player> with his first goal for <team>*
SMT (filled)	na zeventien minuten met tien man verder nadat Jurgen Mattheij met zijn eerste doelpunt voor excelsior
NMT (direct)	tien minuten voor tijd kreeg milot rashica zijn tweede gele kaart . *milot rashica received his second yellow card ten minutes before the end .*
NMT (templatized)	<card_player> kreeg zijn tweede gele kaart . *<card_player> received his second yellow card .*
NMT (filled)	Jurgen Mattheij kreeg zijn tweede gele kaart . *Jurgen Mattheij received his second yellow card .*

Table 5: Examples of the (original and applied) data representation and text output examples for the Dutch Soccer corpus

3.2 Applied methods

Texts and data from the aforementioned datasets served as input and as training (80%), development (10%) and test (10%) set for three forms of trainable data-to-text approaches. Two of them are translation based (NMT and SMT), while the third was a retrieval-based method that served as a baseline. These methods are described in more detail below.

3.2.1 Sentence retrieval

The first method used was a retrieval-based method. Lines from the data representation in the test-set were matched with lines from the data representation in the train-set and assigned a score based on cosine similarity. Cosine similarity scores were obtained by converting the data representation of the target sentence into a bag of words, and doing the same for the data representations in the training set. Subsequently, the (normalized) similarity between the unweighted target data representation and the data representations in the training set is calculated. The line from the train-set with the highest score was chosen and the aligned text sentence was produced as output. A random choice was made between sentences if there were multiple sentences with the highest

[1] `https://github.com/emanjavacas/squawka-scraper`

Corpus	Distor-tion	LM	Word Penalty	Phrase Penalty	Translation Model	Unknown Word Penalty
Weather.gov	0.6	0.8	-1	1e-4	0.6, 1e-4 0.6, 1e-4	2
Prodigy-METEO	0.19	0.69	0	0.29	0.2 0.13, 0.36	0.13
Robocup	0.3	0.5	-1	0.2	0.2, 0.2 0.2, 0.2	0
Dutch Soccer	1e-4	0.8	-3	1e-4	1e-4, 0.6 0.6, 0.6	3

Table 6: MOSES parameters per corpus.

Corpus	Layers	RNN Size	Word Vec Size	Drop-out	Learn-ing Rate	Learning Rate Decay	Batch Size	Beam Size
Weather.gov	1	850	1000	0.15	0.4	0.51	32	5
Prodigy-METEO	1	440	620	0.6	0.4	0.6	1	15
Robocup	1	1230	770	0.39	1	0.6	32	15
Dutch Soccer	2	520	1000	0.15	0.72	0.44	41	14

Table 7: OpenNMT parameters per corpus.

score.

3.2.2 Statistical Machine Translation

The MOSES toolkit (Koehn et al., 2007) was used for SMT. This Statistical Machine Translation system uses Bayes's rule to translate a source language string into a target language string. For this, it needs a translation model and a language model. The translation model was obtained from the parallel corpora described above, while the language model used in the current work is obtained from the text part of the aligned corpora. Translation in the MOSES toolkit is based on a set of heuristics. Parameters of these heuristics were tuned for each corpus using Bayesian Optimization[2] (Snoek et al., 2012). The parameters that returned the highest BLEU score for the non-templatized data were chosen as default parameters for the non-templatized as well as the templatized SMT model. See Table 6 for parameter information.

3.2.3 Neural Machine Translation

Besides Statistical Machine Translation, a Neural Machine Translation approach was explored as well for the current work. These models were trained using the OpenNMT-py toolkit (Klein et al., 2017). Parameters were chosen using the same Bayesian optimization method as was used for SMT. For the smaller corpora (i.e. Prodigy-METEO and Robocup), pre-trained word embeddings were also added to the train model, since these are known to boost performance in low-resource scenarios (Qi et al., 2018). The detailed parameter settings are in Table 7.

4 Templatization and lexicalization

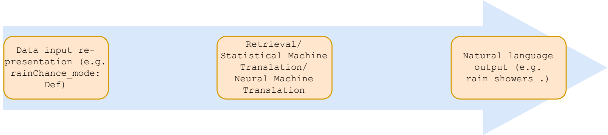
Data input re-presentation (e.g. rainChance_mode: Def)
Retrieval/ Statistical Machine Translation/ Neural Machine Translation
Natural language output (e.g. rain showers .)

Figure 1: Direct method of data-to-text conversion.

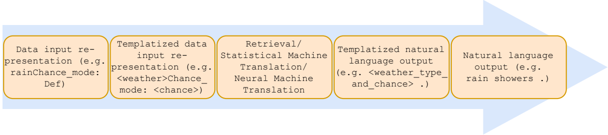
Data input re-presentation (e.g. rainChance_mode: Def)
Templatized data input re-presentation (e.g. <weather>Chance_ mode: <chance>)
Retrieval/ Statistical Machine Translation/ Neural Machine Translation
Templatized natural language output (e.g. <weather_type_ and_chance> .)
Natural language output (e.g. rain showers .)

Figure 2: Templatization method of data-to-text conversion.

The current work investigated differences in output quality for data-to-text generation using 'direct' data-to-text conversion and extended models (see figure 1). For this extended model, the input representation and the text examples in the train and development set were 'templatized'. This means that the natural language sentences were converted to templates by replacing (sets of) words that directly represent (pieces of) data with slots. This replacement was done using a simple set of rules derived from consistencies in the text and data. After this templatization step the data-to-template generation was performed using the methods described in section 3.2, thus generating template sentence texts similar to the ones obtained with the templatization of the text. These obtained templates were finally lexicalized again using similar rules used for the templatization step. Using the original data, gaps were filled with the appropriate information. If multiple options were available to fill the gaps, a weighted random choice was made based on the occurrences of the possibilities in the training set (see figure 2). Thus, after these steps full natural language sentences were created based on a set of (templatized) data.[3]

5 Results automated evaluation

The quality of the generated sentences was assessed using NLTK's *corpus_bleu* that calculates BLEU scores based on 1-grams to 4-grams with equal weights and accounts for a micro-average precision score based on Papineni et al. (2002). Automated metrics such as BLEU have been criticized over the last few years (e.g. Reiter, 2018; Novikova et al., 2017). Especially in the context of NLG. However, Reiter (2018) also suggested that the metric can be used — albeit with caution

[2] https://github.com/fmfn/BayesianOptimization

[3] Code for, and examples of, these steps can be found at https://github.com/TallChris91/Automated-Template-Learning

Corpus	Retrieval Templates (unfilled)	 Templates (filled)	 Direct	SMT Templates (unfilled)	 Templates (filled)	 Direct	NMT Templates (unfilled)	 Templates (filled)	 Direct
Weather.gov	63.94	34.52	69.57	89.29	36.56	61.92	89.85	36.93	78.90
Prodigy-METEO	44.47	27.65	23.66	39.32	26.15	30.37	45.03	26.52	27.82
Robocup	31.39	30.73	22.38	40.77	38.18	39.04	38.98	36.62	37.50
Dutch Soccer	2.49	1.65	4.99	1.64	0.90	2.10	1.95	1.23	1.70

Table 8: BLEU scores obtained for the different corpora with the techniques used in this study.

— for translation tasks, which the current task is in some way. Furthermore, correlations have been found between automated metrics and human ratings (e.g. Belz and Reiter, 2006). Therefore, the BLEU scores were seen as a first step to investigate differences between methods and corpora.

The BLEU scores show that the computer-generated corpora yielded the best results, with Weather.gov showing the best performance compared to the other corpora with BLEU scores for the lexicalized output varying from 34.52 (retrieval using the templatization method) to 78.90 (NMT using the direct method). This seems intuitively logical since the Weather.gov corpus is relatively large, and the sentences are also the most homogeneous out of the corpora, which makes producing output similar to the training data a feasible task. Results for the smaller Robocup soccer corpus are decent, but not as good as Weather.gov with BLEU scores for the lexicalized output ranging from 22.38 (retrieval using the direct method) to 39.04 (SMT using the direct method). While Prodigy-METEO is human-written, its sentence structure is still quite consistent, which might explain why its BLEU scores are not that far removed from those for computer-generated corpora with scores for the lexicalized output between 23.66 (retrieval using the direct method) and 30.37 (SMT using the direct method). Low BLEU scores were obtained for sentences from the Dutch Soccer corpus, with lexicalized output ranging from 0.90 (SMT using the templatization method) to 4.99 (Retrieval using the direct method). The low BLEU scores might indicate two things. First, it is possible that the systems struggle with the heterogeneous nature of the Dutch Soccer texts which results in low text quality output. However, the same heterogeneous nature might also make it difficult to use BLEU scores as an indication for text quality, since it is known to be difficult to find a good gold standard for corpora with diverse language.

BLEU scores for techniques do not show large differences: especially the sentences generated by SMT and NMT obtained close BLEU scores. Interestingly, the sentences produced using cosine similarity based retrieval seems to be consistently outperformed by the translation methods, with the exception of the Dutch Soccer corpus, which suggests that text generation is preferred over simple retrieval. The templatized (filled) and direct methods also scored roughly equal. The exception involves the Weather.gov corpus, where the direct method resulted in much higher BLEU scores compared to its templatized counterpart. Although the results are equal, the metrics show a large decrease in BLEU scores when lexicalizing the templates. This means that the templatization method has the potential to significantly outperform the direct method if the quality of the lexicalization step is improved. See Table 8.

6 Results human evaluation

6.1 Method

Besides an automated metric, a human evaluation was carried out to measure the perceived text quality of sentences from the investigated corpora, techniques and methods. A total of 24 people — all native Dutch students and (junior) colleagues not involved in this research — participated by filling out an online *Qualtrics* survey. Participants were asked to rate sentences generated by the previously described techniques and methods on the aforementioned corpora. For this, a 4 (Corpus: DutchSoccer, Weather.gov, Robocup, Prodigy-METEO) x 3 (Technique: Retrieval, SMT, NMT) x 2 (Method: Templatized, Direct) within-subjects design was implemented. The participants rated 4 sentences per condition — each connected to different data — resulting in a total of 96 sentences that were rated by humans (Krippendorff's α = 0.39; Weighted κ = 0.07).

The participants judged the quality of the sentences on seven-point Likert-scales. These scales measured fluency: how fluent and easy to read the report is ('This text is written in proper Dutch', 'This text is easily readable'), clarity: how clear

	Corpus	Retrieval Templates	Retrieval Direct	SMT Templates	SMT Direct	NMT Templates	NMT Direct
Fluency	Weather.gov	4.08 (1.04)	5.32 (0.88)	5.24 (0.95)	4.76 (0.79)	5.00 (0.97)	5.50 (1.02)
	Prodigy-METEO	3.27 (1.13)	2.81 (1.14)	2.99 (1.16)	3.02 (1.13)	3.31 (1.47)	3.27 (1.43)
	Robocup	5.21 (0.99)	5.46 (1.05)	5.70 (0.99)	4.82 (1.20)	5.59 (1.04)	5.67 (1.11)
	Dutch Soccer	4.12 (0.99)	5.33 (0.91)	2.11 (0.97)	1.78 (0.85)	6.10 (0.84)	5.73 (0.84)
Clarity	Weather.gov	4.36 (1.14)	5.52 (0.99)	5.45 (1.02)	5.24 (1.02)	5.13 (1.26)	5.69 (1.04)
	Prodigy-METEO	2.94 (1.24)	2.73 (1.26)	2.82 (1.27)	2.96 (1.16)	3.25 (1.57)	3.29 (1.47)
	Robocup	5.59 (0.96)	5.73 (1.03)	5.96 (0.92)	5.11 (1.22)	5.84 (0.98)	5.78 (1.37)
	Dutch Soccer	4.85 (1.16)	5.52 (0.90)	2.43 (0.99)	1.94 (0.90)	6.10 (0.92)	5.74 (0.83)
Correctness	Weather.gov	3.34 (0.91)	3.92 (0.90)	2.55 (0.90)	2.70 (1.04)	4.03 (1.04)	3.22 (1.26)
	Prodigy-METEO	4.17 (1.22)	3.21 (0.97)	3.88 (1.23)	3.72 (1.20)	3.99 (1.18)	3.56 (0.88)
	Robocup	5.06 (1.14)	3.83 (1.08)	5.78 (1.08)	5.23 (1.13)	5.70 (1.09)	5.68 (0.92)
	Dutch Soccer	3.34 (0.91)	3.92 (0.90)	2.55 (0.90)	2.70 (1.04)	4.03 (1.04)	3.22 (1.26)

Table 9: Mean fluency, clarity, and correctness scores for the different corpora, techniques and methods. SD is represented between brackets

and understandable the report is ('While reading, I immediately understood the text'), and correctness: how well the information the report is based on is represented in the report itself ('This report does not include extraneous or incorrect information', 'This report does not omit important information'). In order to give ratings on the latter category, participants were provided with a table containing the information used to generate the sentences, followed by six sentences that were generated by the total of six different techniques and methods used in this study. The results were then analyzed using a repeated measures analysis of variance to investigate the effects of the corpus, techniques and methods on text perceptions of fluency, clarity and correctness. Post hoc effects were subsequently measured with a simple effects analysis using the Least Significant Difference test.[4]

6.2 Fluency

For fluency, a main effect was found for corpus ($F(1.89, 43.57) = 56.82$, $p < .001$), as well as technique ($F(2, 46) = 107.13$, $p < .001$), but not for method ($F(1, 23) = 2.22$, $p = .15$). Sentences based on Robocup data resulted in the highest fluency scores ($M = 5.41$, $SD = 0.90$), followed by the Weather.gov corpus ($M = 4.98$, $SD = 0.75$), Dutch Soccer corpus ($M = 4.20$, $SD = 0.50$), and Prodigy-METEO corpus ($M = 3.11$, $SD = 1.12$). Furthermore, sentences generated with NMT generation returned the highest scores on fluency ($M = 5.02$, $SD = 0.76$), followed by Retrieval ($M = 4.45$, $SD = 0.70$), and SMT ($M = 3.80$, $SD = 0.55$) (see table 9).

A significant interaction was also found for corpus x technique ($F(3.07, 70.61) = 87.85$, $p < .001$). NMT resulted in the highest fluency scores for most corpora, except for the Prodigy-METEO corpus where all techniques performed similarly on fluency. A significant interaction was also found for corpus x method ($F(3, 69) = 8.08$, $p < .001$), where the templatization method returned higher fluency scores for the Dutch Soccer and the direct method resulted in higher fluency scores for the Weather.gov corpus. Furthermore, a significant interaction was found for technique x method ($F(2, 46) = 29.76, p < .001$): the fluency scores for the retrieval method were higher when the direct method was used, while the templatization method resulted in higher scores for SMT. A further nuance in this finding can be given with the significant three-way interaction for corpus x technique x method ($F(2.83, 65.08) = 13.89, p < .001$). The templatization method combined with NMT resulted in higher fluency scores for the soccer corpus, but lower scores for the Weather.gov corpus. The same method combined with SMT resulted in higher scores compared to its direct counterpart for all corpora except Prodigy-METEO. For retrieval, the direct method gave higher fluency scores for all corpora.

These scores show that, in general, NMT produces the most fluent sentences. Whether the templatization method or direct method returns the most fluent output depends on the corpus and technique used. For SMT, the templatization method seems the clear winner, but for retrieval and NMT effectiveness of the templatization method differs per corpus. Interestingly, out of all the conditions, the highest fluency scores were obtained for the

[4] Mauchlys Test of Sphericity showed that the sphericity assumption was violated for corpus, corpus x technique, and corpus x technique x method in the case of fluency, as well as clarity. Also for technique, corpus x technique, corpus x method, and technique x method in the case of correctness. Therefore, the Greenhouse-Geisser correction was used for the analyses of these effects.

Dutch Soccer corpus (NMT with the templatization method), while the BLEU scores for this category were fairly low.

6.3 Clarity

The overall scores for clarity look similar to those of fluency. A main effect for corpus was found (F(2.08, 47.72) = 69.90, p < .001), as well as technique (F(2, 46) = 69.21, p < .001), but not for method (F(1, 23) = 1.64, p = .21). Sentences based on Robocup (M = 5.67, SD = 0.89) were considered the clearest, followed by Weather.gov (M = 5.23, SD = 0.89), Dutch Soccer (M = 4.43, SD = 0.48), and Prodigy-METEO (M = 3.00, SD = 1.23). For technique, the lowest clarity scores were found for SMT generated sentences (M = 3.99, SD = 0.61), Retrieval-based sentences (M = 4.66, SD = 0.76) did slightly better, and sentences generated by NMT received the highest clarity scores (M = 5.10, SD = 0.83) (see table 9).[4]

All investigated interactions for clarity were significant (Corpus x technique: F(3.26, 74.89) = 57.936, p < .001; Corpus x method: F(3, 69) = 11.18, p < .001; Technique x method: F(2, 46) = 23.01, p < .001; Corpus x technique x method: F(3.81, 87.56) = 6.03, p < .001). The corpus x technique analysis shows that NMT generated sentences produce the most clear sentences for the Dutch Soccer corpus and the Prodigy-METEO corpus, and NMT and SMT had the shared highest clarity scores for the Weather.gov corpus. No differences in clarity were found for Robocup. Corpus x method results showed no significant difference for the Dutch Soccer and Prodigy-METEO corpus. The direct method resulted in significantly higher scores for the Weather.gov corpus, while sentences generated with the templatization method resulted in higher clarity scores for Robocup sentences. From the technique x method interaction it was observed that Retrieval combined with the direct method resulted in higher clarity scores compared to its templatization counterpart. The opposite is the case for SMT generated sentences, where templatization resulted in higher clarity scores. The three-way interaction of corpus x technique x method showed that NMT produces more clear sentences using the templatization method for the Dutch Soccer corpus and less clear sentences with templatization for the Weather.gov corpus compared to its direct counterpart. Retrieval combined with the direct method scored higher on these corpora with the direct method (vs. templatized), and SMT obtains higher clarity scores for the Dutch Soccer and Robosoccer corpus if the templatization method is applied (vs. templatized).

Overall, models trained on the computer-generated corpora gave the clearest output and, similar to fluency, sentences produced with NMT resulted in the highest clarity scores. Templatization was also overall more effective for SMT compared to the direct method, while templatization for NMT was mostly effective for the Dutch Soccer corpus. The clarity scores for the NMT with templatization method for the Dutch Soccer corpus resulted in the overall highest clarity scores, besides fluency scores as well.

6.4 Correctness

Significant main effects of correctness were found for corpus (F(3, 69) = 32.86, p < .001), technique (F(1.58, 36.37) = 9.25, p = .001), and method (F(1, 23) = 9.77, p = .005). Sentences from the Robocup corpus were deemed the most correct (M = 5.21, SD = 0.92), followed by Weather.gov (M = 4.04, SD = 0.84), with Prodigy-METEO (M = 3.76, SD = 0.88) and Dutch Soccer (M = 3.29, SD = 0.76) in shared last place. For technique, NMT generated sentences were perceived as the most correct (M = 4.27, SD = 0.63). SMT (M = 4.01, SD = 0.72) and Retrieval (M = 3.94, SD = 0.61) did not score significantly different. The results for method showed that templatization resulted in higher correctness scores (M = 4.19, SD = 0.69) than the direct method (M = 3.96, SD = 0.59) (see table 9).

Significant interactions were found for corpus x technique (F(3.64, 83.77) = 20.22, p < .001), corpus x method (F(2.23, 51.29) = 9.24, p < .001), and corpus x technique x method (F(6, 138) = 15.00, p < .001), but not for technique x method (F(1.31, 30.12) = 0.18, p = .84). The corpus x technique interaction shows that SMT generated sentences were perceived as significantly less correct for the Dutch Soccer corpus (vs. Retrieval and NMT), and Retrieval based sentences deemed less correct for Robocup sentences (vs. SMT and NMT). Corpus x method shows that the templatization method resulted in higher perceived correctness for the Robocup and Prodigy-METEO corpora compared to its direct counterpart. Finally, the three way corpus x technique x method interaction shows that templatization combined with NMT resulted in higher correctness scores for Dutch Soccer but lower for Prodigy-METEO (vs. direct). Direct was superior for all corpora when used with a retrieval technique, and the templatization method combined with SMT gives higher scores for the Robocup corpus (vs. direct).

In general, the models trained on the computer-generated corpora produced the most correct sentences. Furthermore, NMT and the templatization method were found to be effective techniques/methods to increase correctness. The fact that templatization increases correctness makes sense since the separate lexicalization step for information ensures that correct information is added to a sentence that is based on the data. This is not necessarily the case with the direct method.

7 Discussion and conclusion

This paper investigated ways to reduce the reliance on rule-based systems when converting data to natural language text. The use of deep learning methods in the form of NMT, and a method where input and output forms were templatized before converting the output template sentences to natural language text were explored. This (relatively) novel NMT approach was compared to more established approaches (i.e. Retrieval and SMT). Furthermore, the templatization method was compared to its direct counterpart that directly converts a data input representation to a natural language text. Sentences were generated for four corpora (two human-written, two computer-generated; two in the sports domain, two in the weather domain). Results of these different forms of generation were then compared using BLEU scores as well as human metrics.

Results of the BLEU scores suggested that the different techniques and approaches obtain the highest text quality when trained on computer-generated corpora, with techniques and approaches trained on the Dutch Soccer corpus generating the lowest text quality output. Furthermore, the Retrieval approach seemed to perform the best in general, and SMT and NMT obtained similar scores to each other. Finally, based on the BLEU scores, the templatization method did not seem to improve output quality when compared to its direct counterpart: similar or higher BLEU scores were found for the direct method.

However, the BLEU results were not corroborated by the results from human evaluation. While the output quality differed per technique, sentences for the Dutch Soccer corpus achieved scores similar or higher than sentences based on other corpora on both fluency, clarity and correctness. Furthermore, the performance of NMT seemed to be good compared to SMT and Retrieval. NMT generated sentences obtained the highest scores on both fluency, clarity and correctness. Also, the templatization method has the potential to increase output quality. Both the SMT and NMT method achieved higher fluency, clarity and correctness scores on sevaral corpora with the templatization method (vs. direct). This method especially seemed to boost performance on the Dutch Soccer corpus: this corpus is the most noisy out of the corpora and contains the most heterogeneous language. Therefore, the templatization method seems to be a useful step for human-written corpora.

The current paper should be seen as a first exploratory step in automating data-to-text systems: the investigated methods could save time and resources compared to a fully rule-based approach, but the steps to templatize data and text for the current article were still rule-based, which still takes manual effort and turned out to decrease output quality based on the BLEU scores. A system that does these conversions automatically would be an interesting avenue for further research. It would also be interesting to extend the current approach to (templated) sentence learning by comparing the translation method to statistical generation techniques such as HMM (e.g. Barzilay and Lee, 2004; Liang et al., 2009) or LSTM (Wen et al., 2015). Other steps in the data-to-text conversion process would be worth investigating as well. For instance automated alignment of data and text, or methods that convert data into the optimal data input representation format, or automated sentence aggregation methods to produce full texts. Further research can also focus on making the output more diverse by adding strategies for lexical variation (Guerini et al., 2011; Gatti et al., 2014). The current results would suggest that combining these steps with the described templatization method, and with NMT, has the potential to further approach the text quality of rule-based systems, and increase overall performance of trainable data-to-text approaches. Especially with noisy human-written corpora containing diverse language.

Acknowledgements

We received support from RAAK-PRO SIA (2014-01-51PRO) and The Netherlands Organization for Scientific Research (NWO 360-89-050), which is gratefully acknowledged. We would also like to thank Abelardo Vieira Mota, and the anonymous reviewers for their comments; Nadine Braun for providing us with the MeMo FC corpus; Chris Emmery, Lieke Gelderloos, and Bram Willemsen for their technical support; and all the people that were willing to participate in the human evaluation study.

References

Ibrahim Adeyanju. 2012. Generating weather forecast texts with case based reasoning. *International Journal of Computer Applications*, 45(10):35–40.

Regina Barzilay and Mirella Lapata. 2005. Collective content selection for concept-to-text generation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP)*, pages 331–338, Vancouver, Canada.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 113–120, Boston, United States.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.

Anja Belz and Eric Kow. 2009. System building cost vs. output quality in data-to-text generation. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 16–24, Athens, Greece.

Anja Belz and Eric Kow. 2010. Assessing the tradeoff between system building cost and output quality in data-to-text generation. In *Empirical methods in natural language generation*, pages 180–200. Springer.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 313–320, Trento, Italy. Association for Computational Linguistics.

Thiago Castro Ferreira, Chris van der Lee, Emiel Krahmer, and Sander Wubben. 2017. Tilburg University models for the WebNLG challenge. In *Proceedings of the 10th International Conference on Natural Language Generation*, Santiago de Compostella, Spain.

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine learning*, pages 128–135, Montreal, Canada.

Kees van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Real vs. template-based Natural Language Generation. *Computational Linguistics*, 31(1):15–23.

Pablo Duboue and Kathleen McKeown. 2002. Content planner construction via evolutionary algorithms and a corpus-based fitness function. In *Proceedings of the International Natural Language Generation Conference*, pages 89–96, Harriman, United States.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostella, Spain.

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*, 61:65–170.

Lorenzo Gatti, Marco Guerini, Oliviero Stock, and Carlo Strapparava. 2014. Sentiment variations in text for persuasion technology. In *International Conference on Persuasive Technology (PERSUASIVE 2014)*.

Dimitra Gkatzia. 2016. Content selection in data-to-text systems: A survey. *arXiv preprint arXiv:1610.08375*.

Eli Goldberg, Norbert Driedger, and Richard I Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Intelligent Systems*, 2:45–53.

Marco Guerini, Carlo Strapparava, and Oliviero Stock. 2011. Slanting existing text with Valentino. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, pages 439–440.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for Neural Machine Translation. *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Philipp Koehn, Hieu Hoang, Alexandra Birch, and Chris Callison-Burch. 2007. MOSES: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180, Prague, Czech Republic.

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*, pages 1406–1415, Sofia, Bulgaria.

Brian Langner, Stephan Vogel, and Alan W Black. 2010. Evaluating a dialog language generation system: Comparing the MOUNTAIN system to other NLG approaches. In *Eleventh Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1109–1112, Makuhari, Japan.

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*, pages 95–104, Santiago de Compostella, Spain.

Chris van der Lee, Bart Verduijn, Emiel Krahmer, and Sander Wubben. 2018. Evaluating the text quality, human likeness and tailoring component of PASS: A Dutch data-to-text system for soccer. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, Santa Fe, United States.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore. Association for Computational Linguistics.

Joy Mahapatra, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. 2016. Statistical natural language generation from tabular non-textual data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 143–152, Edinburgh, Scotland.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2241–2252, Copenhagen, Denmark.

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, Philadelphia, United States.

José Casimiro Pereira, António Teixeira, and Joaquim Sousa Pinto. 2015. Towards a hybrid nlg system for data2text in portuguese. In *Proceedings da 10a Conferência Ibérica de Sistemas e Tecnologias de Informaçao (CISTI)*, pages 679–684, Lisbon, Portugal.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, volume 2, pages 529–535, New Orleans, United States.

Ehud Reiter. 1995. NLG vs. templates. In *Proceedings of the 5th European Workshop on Natural Language Generation (EWNGL)*, pages 95–106, Leiden, The Netherlands.

Ehud Reiter. 2017. You need to understand your corpora! The Weathergov example.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, pages 1–12.

Kyle Richardson, Sina Zarrieß, and Jonas Kuhn. 2017. The code2text challenge: Text generation in source code libraries. In *The 10th International Natural Language Generation conference*, pages 115–119, Santiago de Compostella, Spain.

Lauren Sanby, Ion Todd, and Maria C Keet. 2016. Comparing the template-based approach to GF: the case of Afrikaans. In *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG)*, pages 50–53, Edinburgh, Scotland.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959.

Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2002. Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data. *Technical Report AUCS/TR0201*.

Tsung-Hsien Wen, Milica Gašic, Nikola Mrkšic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1711–1721, Lisbon, Portugal.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2253–2263, Copenhagen, Denmark.

Yuk Wah Wong and Raymond Mooney. 2007. Generation by inverting a semantic parser that uses statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference (NAACL HLT)*, pages 172–179, Rochester, USA.