The WebNLG Challenge: Generating Text from RDF Data

Claire Gardent **Anastasia Shimorina**
CNRS, LORIA, UMR 7503
Vandoeuvre-lès-Nancy, F-54500, France
claire.gardent@loria.fr
anastasia.shimorina@loria.fr

Shashi Narayan **Laura Perez-Beltrachini**
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh, EH8 9AB, UK
shashi.narayan@ed.ac.uk
lperez@ed.ac.uk

Abstract

The WebNLG challenge consists in mapping sets of RDF triples to text. It provides a common benchmark on which to train, evaluate and compare "microplanners", i.e. generation systems that verbalise a given content by making a range of complex interacting choices including referring expression generation, aggregation, lexicalisation, surface realisation and sentence segmentation. In this paper, we introduce the microplanning task, describe data preparation, introduce our evaluation methodology, analyse participant results and provide a brief description of the participating systems.

1 Introduction

Previous Natural Language Generation (NLG) challenges have focused on surface realisation (Banik et al., 2013; Belz et al., 2011), referring expression generation (Belz and Gatt, 2007; Gatt et al., 2008; Gatt et al., 2009; Belz et al., 2008; Belz et al., 2009; Belz et al., 2010) and content selection (Bouayad-Agha et al., 2013).

In contrast, the WebNLG challenge focuses on microplanning, that subtask of NLG which consists in mapping a given content to a text verbalising this content. Microplanning is a complex choice problem involving several subtasks referred to in the literature as referring expression generation, aggregation, lexicalisation, surface realisation and sentence segmentation. For instance, given the WebNLG data unit shown in (1a), generating the text in (1b) involves choosing to lexicalise the JOHN_E_BLAHA entity only once (*referring expression generation*), lexicalising the OCCUPATION property as the phrase *worked as* (*lexicalisation*), using PP coordination to avoid repeating the word born (*aggregation*) and verbalising the three triples by a single complex sentence including an apposition, a PP coordination and a transitive verb construction (*sentence segmentation* and *surface realisation*).

(1) a. Data: (JOHN_E_BLAHA BIRTHDATE 1942_08_26) (JOHN_E_BLAHA BIRTHPLACE SAN_ANTONIO) (JOHN_E_BLAHA OCCUPATION FIGHTER_PILOT)

b. Text: *John E Blaha, born in San Antonio on 1942-08-26, worked as a fighter pilot.*

2 Data

As illustrated by the above example, the WebNLG dataset was designed to exercise the ability of NLG systems to handle the whole range of microplanning operations and their interactions. It was created using a content selection procedure specifically designed to enhance data and text variety (Perez-Beltrachini et al., 2016). In (Gardent et al., 2017), we compared a dataset created using the WebNLG process with existing benchmarks in particular, (Wen et al., 2016)'s dataset (RNNLG) which was produced using a similar process. In what follows, we give various statistics about the WebNLG dataset using the RNNLG dataset as a reference point.

Size. The WebNLG dataset consists of 25,298 (data,text) pairs and 9,674 distinct data units. The data units are sets of RDF triples extracted from DBpedia and the texts are sequences of one or more sentences verbalising these data units.

Lexicalisation. As illustrated by the examples in (2), different properties can induce different lexical forms (a property might be lexicalised as a verb, a relational noun, a preposition or an adjective). Therefore, the larger the number of properties, the more likely the data is to allow for a wider range of lexicalisation patterns.

(2) X TITLE Y ⇒ *X served as Y* — Verb
X NATIONALITY Y ⇒ *X's nationality is Y* — Relational noun
X COUNTRY Y ⇒ *X is in Y* — Preposition
X NATIONALITY USA ⇒ *X is American* — Adjective

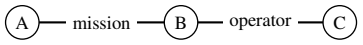
To promote diverse lexicalisation patterns, we extracted data from 15 DBpedia categories (Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, WrittenWork, Athlete, Artist, City, MeanOfTransportation, CelestialBody, Politician) resulting in a set of 373 distinct RDF properties (more than three times the number of properties contained in the RNNLG dataset). The corrected type token ratio (CTTR[1]) and the number of word types is roughly twice as large in the WebNLG dataset than in RNNLG.

Surface Realisation. To increase *syntactic variety*, we use a content selection procedure which extracts data units of various shapes. The intuition is that different input shapes may induce distinct linguistic constructions. This is illustrated in Figure 1. Typically, while triples sharing a subject (SIBLING configuration) are likely to induce a VP or a sentence coordination, a CHAIN configuration (where the object of one triple is the subject of the other) will more naturally give rise to object relative clauses or participials.

Another factor impacting syntactic variation is the set of properties (input patterns) cooccurring in a given input. This is illustrated by the examples in (3) where two inputs of the same length (3 triples hence 3 properties) result in text with different syntax. That is, a larger number of input patterns is more likely to induce texts with greater syntactic variety. By extracting data units from a large number of distinct input patterns.

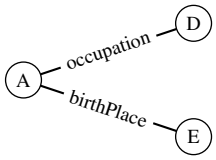
[1]Following (Perez-Beltrachini and Gardent, 2017), we use (Lu, 2008)'s system to compute the CTTR (Carroll, 1964).

CHAIN



*A participated in **C operated B** mission*

SIBLING



*A was born in E. **She** worked as a D.*
*A was born in E **and** worked as a D.*

Figure 1: Input shapes and linguistic structures.

(3) a. LOCATION-COUNTRY-STARTDATE
⇒ Passive-Apposition-Active
108 St. Georges Terrace is located in Perth, Australia. Its construction began in 1981.

b. BIRTHPLACE-ALMAMATER-SELECTION
⇒ Passive-VP coordination
William Anders was born in British Hong Kong, graduated from AFIT in 1962, and joined NASA in 1963.

As shown in Table 1, the WebNLG dataset contains twice as many distinct input patterns and ten times more input shapes than the RNNLG dataset. It is also less redundant with a ratio between number of inputs and number of input patterns of 2.34 against 10.31 for RNNLG.

Aggregation, Sentence Segmentation and Referring Expression Generation. Finally, the need for aggregation, sentence segmentation and referring expression generation mainly arises when texts contain more than one sentence. As Table 1 shows, although data units are overall smaller in the WebNLG dataset than in RNNLG, the WebNLG dataset has a higher number of texts containing more than one sentence and contains texts of longer length.

3 Participating Systems

The WebNLG challenge received eight submissions from six participating teams: the ADAPT Centre, Ireland (ADAPTCENTRE), the University of Melbourne, Australia (UMELBOURNE), Peking University, China (PKUWRITER), Tilburg University, The

	WebNLG	RNNLG
Size		
# data-text pairs	25,298	30,842
# distinct inputs	9,674	22,225
Lexicalisation		
# properties	373	108
# domains	15	4
# CTTR	6.51	3.42
# Words (Type)	6,547	3,524
Syntactic Variety		
# input patterns	4,129	2,155
# input / # input patterns	2.34	10.31
# input shapes	62	6
Aggregation, GRE, Segmentation		
# input with 1 or 2 triples	11,111	4,087
# input with 3 or 4 triples	8,172	6,690
# input with 5 to 7 triples	6,015	20,065
# text with 1 sentence	16,740	24,234
# text with 2 sentences	6,798	5,729
# text with $\geq$ 3 sentences	1,760	879
# words/text (avg/min/max)	22.69/4/80	18.37/1/76

Table 1: Some statistics about the WebNLG dataset.

Netherlands (UTILBURG), University of Information Technology, VNU-HCM, Vietnam (UIT-VNU-HCM) and Universitat Pompeu Fabra, Barcelona, Spain (UPF-FORGE). Each team submitted outputs from a single system except UTILBURG who submitted outputs from three different systems. As a result, there were nine systems in total: eight participating systems and our baseline (BASELINE) system. These can be grouped into three categories: pipeline systems, statistical machine translation (SMT) and neural machine translation (NMT) systems. Table 2 shows the system categorisations.

System ID	Institution
PIPELINE Systems	
UTILBURG-SMT	Tilburg University
UIT-VNU-HCM	University of Information Technology
UPF-FORGE	Universitat Pompeu Fabra
SMT Systems	
UTILBURG-SMT	Tilburg University
NMT Systems	
ADAPTCENTRE	ADAPT Centre, Ireland
UMELBOURNE	University of Melbourne
UTILBURG-NMT	Tilburg University
PKUWRITER	Peking University
BASELINE	

Table 2: Categorisation of participating systems.

Pipeline Systems. Three submissions used a template or grammar-based pipeline framework with some NLG module: UTILBURG-PIPELINE, UIT-VNU-HCM and UPF-FORGE.

The first two systems, UTILBURG-PIPELINE and UIT-VNU-HCM, extracted rules or templates from the training data for surface realisation, whereas the third system, UPF-FORGE, used the FORGe grammar (Mille et al., 2017).

UTILBURG-PIPELINE extracted rules mapping a triple (or a triple set) to a text observed in the training data; both the triple and the associated text were delexicalised. Given an RDF triple set to generate from, UTILBURG-PIPELINE first ordered triples to maintain discourse order. Extracted rules were then applied to generate a delexicalised text. Missing entities were added using a referring expression generation module (Castro Ferreira et al., 2016). Finally,

a 6-gram language model trained on the Gigaword corpus was used to rank the system output.

UIT-VNU-HCM did not resort to delexicalisation in their rules. Instead of using the text to extract templates, it used the typed-dependency structure of the text to facilitate rule extraction from the training data. In addition, at run time, WordNet was used to estimate similarity between predicates in the test and train sets.

UPF-FORGE mostly focused on sentence planning with predicate-argument (PredArg) templates. For each of the DBpedia properties found in the training and evaluation data, they manually defined PredArg templates encoding various DBpedia-specific and linguistic features. Given an RDF triple set to generate from, PredArg templates were used to convert these triples to PredArg structures and to further aggregate them to form a PredArg graph structure. The FORGe generator took this linguistic PredArg structure as input and generated a text.

SMT Systems. UTILBURG-SMT was the only system which used the statistical machine translation framework. It was trained on the WebNLG dataset using the Moses toolkit (Koehn et al., 2007). The dataset was pre-processed whereby each entity in the input and each corresponding referring expression in the output were delexicalised and annotated with the entity Wikipedia ID. The alignments from the training set were obtained using MGIZA and model weights were tuned using 60-batch MIRA with BLEU as the evaluation metric. Similar to UTILBURG-PIPELINE, the system used a 6-gram language model trained on the Gigaword corpus using KenLM.

NMT Systems. Four systems build upon the attention-based encoder-decoder architecture proposed by the machine translation community (Bahdanau et al., 2014). These systems are: ADAPTCENTRE, UMELBOURNE, UTILBURG-NMT and PKUWRITER. Indeed, most of them make use of existing NMT frameworks. There are however important differences among the four systems with respect to the concrete architecture and sequence representations used.

ADAPTCENTRE makes use of the Nematus (Sennrich et al., 2017) system. They opt for subword representations rather than delexicalisation to deal with rare words and sparsity. They linearise the input sequence and insert tuple separation special tokens.

UMELBOURNE does a combined delexicalisation procedure and enrichment of the input sequence. Entities are delexicalised using an entity identifier (ENTITY-ID) and the DBpedia type of the entity, when available, is also appended to the input sequence. An n-gram search is used to assure the most accurate target sequence delexicalisation. They use a standard encoder-decoder with attention model.

UTILBURG-NMT is based on the Edinburgh Neural Machine Translation submission for the 2016 machine translation shared task (WMT 2016). The target sequences are the delexicalised texts (cf. UTILBURG-PIPELINE) and the input sequences are the linearisation of the delexicalised input set of triples. The REG module from their pipeline system is used to post-process the decoder outputs.

The PKUWRITER system relies upon two extra mechanisms, namely a ranking module and an extra Reinforcement Learning (RL) training objective. It uses an ensemble of attention-based encoder-decoder models based on the TensorFlow seq2seq API in addition to the baseline (7 models in total). They propose an output ranking module to choose the best verbalisation among those outputs by the generation models. The ranker is trained on supervised data generated automatically. Input triple sets are paired with verbalisations produced by each of the generation models. Then, each pair is associated with a quality score, i.e. the BLEU score of the verbalisation and the reference. Word and sentence level features are extracted to train the ranker. The generation models and ranker are trained on different dataset partitions. The RL objective aims at encouraging the generator to include triple subjects in the produced verbalisation. Finally, PKUWRITER uses a set of hand-crafted rules to handle input cases where the model fails.

4 Evaluation Methodology

The WebNLG challenge includes both an automatic and a human-based evaluation. Due to time constraints, only the results of the automatic evaluation are presented in this paper. The results of the human-

based evaluation will be provided on the WebNLG website[2] in October 2017.

4.1 Automatic Evaluation

Three automatic metrics were used to evaluate the participating systems:

- BLEU-4[3] (Papineni et al., 2002). BLEU scores were computed using up to three references.
- METEOR (v1.5)[4] (Denkowski and Lavie, 2014);
- TER[5] (Snover et al., 2006).

For statistical significance testing, we followed the bootstrapping algorithm described in (Koehn and Monz, 2006).

To assess the ability of the participating system to generalise from out-of-domain data, the test dataset consists of two sets of roughly equal size: a test set containing inputs created for entities belonging to DBpedia categories that were seen in the training data (Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, City, and WrittenWork), and a test set containing inputs extracted for entities belonging to 5 unseen categories (Athlete, Artist, MeanOfTransportation, CelestialBody, Politician). We call the first type of data *seen categories*, the second, *unseen categories*. Correspondingly, we report results for 3 datasets: the seen category dataset, the unseen category dataset and the total test set including both data from seen and from unseen categories.

Table 4.1 gives more detailed statistics about the number of properties, objects and subject entities which occur in each test set.

- $|Test|$ is the number of distinct properties / subjects / objects in the test set;
- $|Test \cap TnDv|$ is the number of distinct properties / subjects / objects which are in the test set and were seen in the training or the development set;
- $|Test \setminus TnDv|$ is the number of distinct properties / subjects / objects which are in the test set, but not in the training and development set.

		Seen	**Unseen**	**All**
Prop.	$\|Test\|$	188	159	300
	$\|Test \cap TnDv\|$	188	51	192
	$\|Test \setminus TnDv\|$	0	108	108
Obj.	$\|Test\|$	1033	898	1888
	$\|Test \cap TnDv\|$	1011	57	1025
	$\|Test \setminus TnDv\|$	22	841	863
Subj.	$\|Test\|$	343	238	575
	$\|Test \cap TnDv\|$	342	6	342
	$\|Test \setminus TnDv\|$	1	232	233

Table 3: Test data statistics on properties, objects and subjects for seen, unseen and all datasets.

While in the seen test data (first column) almost all triple elements are present in the training and development sets, in the unseen test data (second column) the vast majority of subjects, objects, and, more importantly, properties (which need to be verbalised) has not been seen in the training and development data.

Participants were requested to submit tokenised and lowercased texts. To ensure consistency between submissions, we pre-process the submitted results one more time to double-check that those requirements were fullfilled. As teams used different strategies of tokenisation, we had to modify submissions using our own scripts. In particular, all punctuation signs were separated from alphanumeric sequences (e.g. a two-token group *65.6 feet* was modified to a four-token *65 . 6 feet*). Moreover, we converted both references and submission outputs to the ASCII character set.

4.2 Baseline System

We developed a baseline system using neural networks and delexicalisation. Before training, we preprocess the data by linearising triples, performing tokenisation and delexicalisation using exact matching.

While delexicalising, we make the following replacements:

- given a triple of the form *(s p o)* where s is of the category C for which the triple set has been

[2] http://talc1.loria.fr/webnlg/stories/challenge.html

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[4] http://www.cs.cmu.edu/~alavie/METEOR/

[5] http://www.cs.umd.edu/~snover/tercom/

produced (e.g., *Alan_Bean* for the category Astronaut), we replace s by C.

- given a triple of the form *s p o*, we replace o by p. E.g., *(s country Indonesia)* becomes (*s country COUNTRY*). The replacements were made using the exact match, in such a way not all the entities were replaced.

Examples 4 and 5 show a (data,text) pair before and after delexicalisation. Note that *noodles* was not substituted by the corresponding entity category in the target text (because there is no exact match with the NOODLE object in the input). Table 4 shows the number of distinct tokens occurring in the original and delexicalised data.

(4) a. Set of triples: (INDONESIA LEADERNAME JUSUF_KALLA) (BAKSO INGREDIENT NOODLE) (BAKSO COUNTRY INDONESIA)

b. Text: *Bakso is a food containing noodles; it is found in Indonesia where Jusuf Kalla is the leader.*

(5) a. Source: (COUNTRY LEADERNAME LEADERNAME) (FOOD INGREDIENT INGREDIENT) (FOOD COUNTRY COUNTRY)

b. Target: *FOOD is a food containing noodles ; it is found in COUNTRY where LEADERNAME is the leader .*

On this delexicalised data-to-text corpus, we trained a vanilla sequence-to-sequence model with attention mechanism using the OpenNMT toolkit (Klein et al., 2017) with default parameters for training and translating. The network consists of a two-layered bidirectional encoder-decoder model with LSTM units. We use a batch size of 64 and a starting learning rate of 1.0. The size of the hidden states is 500. The network was trained for 13 epochs with a stochastic gradient descent optimisation method and a dropout probability of 0.3. We used the entire vocabulary for the baseline due to its rather small size.

After training we relexicalised sentences with corresponding entities if of course their counterparts were present in generated output. The performance of the baseline is shown in Tables 5, 6, 7 along with other teams' results.

	Original	**Delexicalised**
Source	2703	1300
Target	5374	5013
Total	8077	6313

Table 4: Vocabulary size in tokens.

5 Results

We briefly discuss the automatic scores distinguishing between results on the whole dataset, on data extracted from previously unseen categories and on data extracted from seen categories.

Global Scores. Table 5 shows the global results that is, the results for the whole test set. Horizontal lines group together systems for which the difference in scores is not statistically significant. The names of the teams are coloured according to the system type they explored: neural-based systems are in red, pipeline systems in blue, and SMT systems in light grey.

Most systems (6 out of 8) outperform the baseline, four of them obtaining scores well above it. In terms of BLEU and TER scores, the four first systems include systems of each type (neural, SMT-based and pipelines).

While BLEU and TER yield almost identical rankings, METEOR does not, suggesting that the systems handle synonyms and morphological variation differently. In particular, the fact that UPF-FORGE ranks first under the METEOR score suggests that it often generates text that differs from the references because of synonymic or morphological variation.

Scores on Seen Categories. For data extracted from DBpedia categories that were seen in the training data, machine learning based systems (neural and SMT) mostly outperform more rule-based systems. Notably, in terms of BLEU and TER scores, the three pipeline systems are at the end of the rating. Again though, the METEOR scores show a much higher ranking (3rd rather than 6th) for the UPF-FORGE system.

Scores on Unseen Categories. On unseen categories, the UPF-FORGE systems ranks first as the system could quickly be adapted to handle properties that had not been seen in the training data. The

BLEU		
1	MELBOURNE	45.13
2	TILB-SMT	44.28
3–4	PKUWRITER	39.88
3–4	UPF-FORGE	38.65
5–6	TILB-PIPELINE	35.29
5–6	TILB-NMT	34.60
7	BASELINE	33.24
8	ADAPT	31.06
9	UIT-VNU	7.07

TER		
1	MELBOURNE	0.47
2	TILB-SMT	0.53
3–4	PKUWRITER	0.55
3–5	UPF-FORGE	0.55
4–5	TILB-PIPELINE	0.56
6–7	TILB-NMT	0.60
6–7	BASELINE	0.61
8–9	UIT-VNU	0.82
8–9	ADAPT	0.84

METEOR		
1	UPF-FORGE	0.39
2	TILB-SMT	0.38
3	MELBOURNE	0.37
4	TILB-NMT	0.34
5–6	ADAPT	0.31
5–7	PKUWRITER	0.31
6–7	TILB-PIPELINE	0.30
8	BASELINE	0.23
9	UIT-VNU	0.09

Table 5: Results for all categories. Lines between systems indicate a difference in scores which is statistically significant ($p < 0.05$). A colour for a team name indicates a type of the system used (NMT, SMT, Pipeline).

BLEU		
1	ADAPT	60.59
2–3	MELBOURNE	54.52
2–4	TILB-SMT	54.29
3–4	BASELINE	52.39
5	PKUWRITER	51.23
6	TILB-PIPELINE	44.34
7	TILB-NMT	43.28
8	UPF-FORGE	40.88
9	UIT-VNU	19.87

TER		
1	ADAPT	0.37
2	MELBOURNE	0.40
3–4	BASELINE	0.44
3–4	PKUWRITER	0.45
5	TILB-SMT	0.47
6	TILB-PIPELINE	0.48
7	TILB-NMT	0.51
8	UPF-FORGE	0.55
9	UIT-VNU	0.78

METEOR		
1	ADAPT	0.44
2	TILB-SMT	0.42
3–4	MELBOURNE	0.41
3–4	UPF-FORGE	0.40
5–6	TILB-NMT	0.38
5–8	TILB-PIPELINE	0.38
6–8	PKUWRITER	0.37
6–8	BASELINE	0.37
9	UIT-VNU	0.15

Table 6: Results for seen categories.

BLEU		
1	UPF-FORGE	35.70
2	MELBOURNE	33.27
3	TILB-SMT	29.88
4–5	PKUWRITER	25.36
4–5	TILB-NMT	25.12
6	TILB-PIPELINE	20.65
7	ADAPT	10.53
8	BASELINE	06.13
9	UIT-VNU	0.11

TER		
1	UPF-FORGE	0.55
2	MELBOURNE	0.55
3	TILB-SMT	0.61
4–5	TILB-PIPELINE	0.65
4–5	PKUWRITER	0.67
6	TILB-NMT	0.72
7	BASELINE	0.80
8	UIT-VNU	0.87
9	ADAPT	1.4

METEOR		
1	UPF-FORGE	0.37
2	TILB-SMT	0.33
3	MELBOURNE	0.33
4	TILB-NMT	0.31
5	PKUWRITER	0.24
6	TILB-PIPELINE	0.21
7	ADAPT	0.19
8	BASELINE	0.07
9	UIT-VNU	0.03

Table 7: Results for unseen categories.

S John Clancy is a labour politican who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.

$\mathbf{M}_S$ { BIRMINGHAM|LEADERNAME|JOHN_CLANCY_(LABOUR_POLITICIAN), JOHN_MADIN|BIRTHPLACE|BIRMINGHAM, 103_COLMORE_ROW|ARCHITECT|JOHN_MADIN}

$\mathbf{T}_1$ Labour politician, John Clancy is the leader of Birmingham.

$\mathbf{M}_{T_1}$ { BIRMINGHAM|LEADERNAME|JOHN_CLANCY_(LABOUR_POLITICIAN)}

$\mathbf{T}_2$ John Madin was born in Birmingham.

$\mathbf{M}_{T_2}$ { JOHN_MADIN|BIRTHPLACE|BIRMINGHAM }

$\mathbf{T}_3$ He was the architect of 103 Colmore Row.

$\mathbf{M}_{T_3}$ { 103_COLMORE_ROW|ARCHITECT|JOHN_MADIN}

Figure 2: An example pair out of the Split-and-Rephrase Dataset. **S** is a single complex sentence with meaning $\mathbf{M}_S$. $\mathbf{T}_1$, $\mathbf{T}_1$, $\mathbf{T}_1$ form a text of three simple sentences whose joint meaning $\mathbf{M}_{T_1} \cup \mathbf{M}_{T_2} \cup \mathbf{M}_{T_3}$ is the same as the meaning $\mathbf{M}_S$ of the corresponding single complex sentence **S**.

ranking of the other systems is more or less unchanged with the exception of the ADAPTCENTRE system. This neural system does not use delexicalisation and the subword approach that was adopted to handle unseen data does not seem to work well.

6 Conclusion

The WebNLG challenge was novel in that it was the first challenge to provide a benchmark on which to evaluate and compare microplanners. Despite a tight schedule (we released the training data in April for a submission in August), it generated a high level of interest among the NLG community: 62 groups from 18 countries[6] downloaded the data, 6 groups submitted 8 systems and 3 groups developed a system but did not submit.

The training data for the WebNLG 2017 challenge is available on the WebNLG website[7] and evaluation on the test data can be run by the organisers on demand. A larger dataset consisting of 40,049 (data, text) pairs, 15,095 distinct data input and 15 DBpedia categories is also available. Both datasets are under the Creative Commons license "CC Attribution-Noncommercial-Share Alike 4.0 International license". We hope that these resources will enable a long and fruitful strand of research on microplanning.

The usefulness of the WebNLG dataset reaches far beyond the WebNLG challenge. It can be used for instance to train a semantic parser which would convert a sentence into a set of RDF triples. It can also be used to derive new datasets for related tasks. Thus in (Narayan et al., 2017), we show how to derive from the WebNLG dataset a dataset for sentence simplification which we call the Split-and-Rephrase dataset. In this dataset, each pair consists of (i) a single, complex sentence and its meaning representation in terms of RDF triples and (ii) a sequence of at least two sentences and their corresponding RDF triples. In other words, the Split-and-Rephrase dataset associates a complex sentence with a sequence of at least two sentences whose meaning is the same as that of the complex sentence. As explained in (Narayan et al., 2017), this dataset was created using the meaning representations (sets of RDF triples) as pivot. The Split-and-Rephrase dataset consists of 1,100,166 pairs of the form $\langle (M_C, T_C), \{(M_1, T_1) \dots (M_n, T_n)\}\rangle$ where T_C is a complex sentence and $T_1 \dots T_n$ is a sequence of texts with semantics $M_1, \dots M_n$ expressing the same content M_C as T_C. Figure 2 shows an example pair. It was used to train four neural systems, and the associated meaning representations were shown to improve performance.

In the future, we are planning to build a multilingual resource in which the English text present in the WebNLG dataset will be translated into French, Russian and Maltese. In this way, morphological variation can be explored which is an interesting avenue of research in particular for neural systems which have a limited ability to handle unseen input: how well will these systems be able to handle the generation of morphologically rich languages?

The analysis of the participants' results presented in this paper will be complemented in an arXiv report by the results of a human-based evaluation. Using human judgements obtained through crowdsourcing, this human evaluation will assess the system results on three criteria, namely fluency, grammaticality and appropriateness (does the text correctly verbalise the input data?). We will also provide a more in depth analysis of the participant results on data extracted from different categories and data of various length.

[6] Australia, Canada, China, Croatia, France, Germany, India, Iran, Ireland, Italy, Netherlands, Norway, Poland, Spain, Tunisia, UK, USA, Vietnam

[7] http://talc1.loria.fr/webnlg/stories/challenge.html

Acknowledgments

The research presented in this paper has been supported by the following grants and projects: "WebNLG", Project ANR-14-CE24-0033 of the French National Research Agency and "SUMMA", H2020 project No. 688139.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR-2015 (abs/1409.0473)*.

Eva Banik, Claire Gardent, and Eric Kow. 2013. The kbgen challenge. In *the 14th European Workshop on Natural Language Generation (ENLG)*, pages 94–97.

Anja Belz and Albert Gatt. 2007. The attribute selection for gre challenge: Overview and evaluation results. *Proceedings of UCNLG+ MT: Language Generation and Machine Translation*, pages 75–83.

Anja Belz, Eric Kow, Jette Viethen, and Albert Gatt. 2008. The grec challenge: Overview and evaluation results.

Anja Belz, Eric Kow, and Jette Viethen. 2009. The grec named entity generation challenge 2009: overview and evaluation results. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*, pages 88–98. Association for Computational Linguistics.

Anja Belz, Eric Kow, Jette Viethen, and Albert Gatt. 2010. Generating referring expressions in context: The grec task evaluation challenges. In *Empirical methods in natural language generation*, pages 294–327. Springer.

Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the 13th European Workshop on Natural Language Generation*, ENLG '11, pages 217–226, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nadjet Bouayad-Agha, Gerard Casamayor, Leo Wanner, and Chris Mellish. 2013. Overview of the first content selection challenge from open semantic web data. In *ENLG*, pages 98–102.

J. B. Carroll. 1964. *Language and thought.* NJ: Prentice-Hall. Englewood Cliffs.

Thiago Castro Ferreira, Emiel Krahmer, and Sander Wubben. 2016. Towards more variation in text generation: Developing and evaluating variation models for choice of referential form. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 568–577.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 179–188.

Albert Gatt, Anja Belz, and Eric Kow. 2008. The tuna challenge 2008: Overview and evaluation results. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 198–206. Association for Computational Linguistics.

Albert Gatt, Anja Belz, and Eric Kow. 2009. The tuna-reg challenge 2009: Overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 174–182. Association for Computational Linguistics.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '06, pages 102–121.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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.

Xiaofei Lu. 2008. Automatic measurement of syntactic complexity using the revised developmental level scale. In *FLAIRS Conference*, pages 153–158.

Simon Mille, Roberto Carlini, Alicia Burga, and Leo Wanner. 2017. FORGe at SemEval-2017 task 9: Deep sentence generation based on a sequence of graph transducers. In *Proceedings of SemEval-2017*, pages 917–920.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of EMNLP*.

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

Laura Perez-Beltrachini and Claire Gardent. 2017. Analysing data-to-text generation benchmarks. In *Proceedings of the tenth International Natural Language Generation Conference*, INLG.

Laura Perez-Beltrachini, Rania Sayed, and Claire Gardent. 2016. Building rdf content for data-to-text generation. In *COLING*, pages 1493–1502.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch-Mayne, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Laubli, Antonio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: A toolkit for neural machine translation. In *Proceedings of the EACL 2017 Software Demonstrations*, pages 65–68, 4.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of

translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of NAACL-HLT*.