DCU/TCD-FORGe at WebNLG'23: Irish rules!

Simon Mille
ADAPT, Dublin City University
simon.mille@adaptcentre.ie

Elaine Uí Dhonnchadha
Trinity College, Dublin
uidhonne@tcd.ie

Stamatia Dasiopoulou
Independent Researcher
stamatia.dasiopoulou@gmail.com

Lauren Cassidy
ADAPT, Dublin City University
lauren.cassidy@adaptcentre.ie

Brian Davis
ADAPT, Dublin City University
brian.davis@adaptcentre.ie

Anya Belz
ADAPT, Dublin City University
anya.belz@adaptcentre.ie

Abstract

In this paper, we describe the submission of Dublin City University (DCU) and Trinity College Dublin (TCD) for the WebNLG 2023 shared task. We present a fully rule-based pipeline for generating Irish texts from DBpedia triple sets which comprises 4 components: triple lexicalisation, generation of non-inflected Irish text, inflection generation, and post-processing. Our whole pipeline is available at https://github.com/mille-s/DCU_TCD-FORGe_WebNLG23

1 Introduction

The WebNLG dataset is a benchmark for data-to-text Natural Language Generation (NLG) consisting of {input, output} pairs, where the input is a set of n triples ($1 \leq n \leq 7$) and the output a set of m texts that verbalise each triple set. The triples are extracted from DBpedia and represent relationships between DBpedia resources, namely subjects (*DB-Subj*) and objects (*DB-Obj*), via respective properties e.g. for the property *country*: `Texas | country | United_States`. The WebNLG'23 shared task focuses on four languages for which the existing resources are limited: Irish, Welsh, Breton and Maltese; we submitted Irish outputs only.

Large Language Models are becoming more and more used for NLG, but it is well known that they are heavily dependent on the quantity and quality of data they are trained on. On the other hand, rule-based systems although limited in terms of coverage and/or fluency are usually easier to adapt to low-resource languages. For our DCU/TCD-FORGe submission, our pipeline consists of 4 fully rule-based modules, which are described in the remainder of the paper: **Lexicalisation of input triples**: instantiation of predicate-argument templates with DB-Subj and DB-Obj values (Section 2); **Generation of non-inflected Irish sentences**: a sequence of graph transducers that progressively specify the linguistic structures into their surface-oriented form (FORGe, Section 3); **Generation of morphological inflections**: finite-state transducers to produce inflections via two-level morphology (Section 4); and **Post-processing**: cleaning and formatting of the text (Section 5). DCU/TCD-FORGe uses disk space of ∼8MB and runs with less than 1GB of RAM; it generates the WebNLG test set (1,779 texts) in ∼15 min (∼0.5 sec/text) and achieved 0.167 BLEU.

2 Lexicalisation of Input Triples

In this section, we detail the steps involved in the lexicalisation of the WebNLG inputs: we start with the predicate-argument (PredArg) templates and their instantiation (2.1), and continue with the adaptations carried out for Irish generation (2.2).

2.1 PredArg templates and their instantiation

For lexicalisation, we follow the approach of the FORGe submission at WebNLG'19 (Mille et al., 2019), i.e. we use PredArg templates in the PropBank style (Kingsbury and Palmer, 2002) that correspond to each individual property and instantiate them by replacing the DB-Subj and DB-Obj placeholders with their respective lexicalisations (see Section 2.2). The instantiated templates are then grouped based on their DB-Subj and ordered in descending frequency of appearance of the DB-Subj

in the input triple set (e.g. the triples with a DB-Subj that has 3 mentions come before those with 2 mentions). Figure 1 shows a PredArg template, instantiated in Figure 3 in Appendix A.

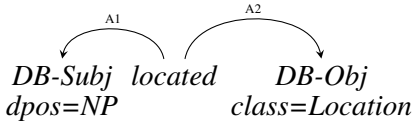

Figure 1: Sample PredArg template corresponding to the *country* property.

2.2 Adaptations for Irish

Lexicalisation of properties. We handcrafted templates for all properties already covered by FORGe, i.e. the training, development and test properties of WebNLG'20 (Castro Ferreira et al., 2020); there were no new properties in the 2023 dataset, so our generator is able to generate all the 2023 inputs. There are 411 different properties, and since several properties can be verbalised the same way,[1] the total number of unique templates is lower (381).

Lexicalisation of DB-Subj and DB-Obj values. For each triple, the property and its pertinent domain and range classes determine whether the DB-Subj and DB-Obj values will be lexicalised using their English or Irish label (human readable name). To obtain the latter, we take advantage of the *owl:sameAs* relation that links the DB-Subj (DB-Obj) entity of the English DBpedia to its equivalent entity in the Irish DBpedia version; if no equivalent entity is contained in the localised DBpedia version, we fall back to Google translate,[2] giving as input the English label without any further context.

3 Generation of non-inflected sentences

In this section, we describe FORGe (3.1) and provide an overview of its extensions for Irish (3.2).

3.1 FORGe

FORGe (Mille et al., 2019) is a rule-based generator that takes as input minimal PredArg structures. It realises the last four consecutive steps of the traditional NLG pipeline (Reiter and Dale, 1997) (sentence aggregation, lexicalisation,[3] referring expression generation and linguistic realisation), except that for WebNLG'23, the output of FORGe is a sequence of lemmas with morphological information instead of inflected words (see Section 4). Each of the four steps is implemented as one or more graph transducer(s) that successively map the input PredArg onto different dependency-based intermediate linguistic representations, loosely following the different levels of Meaning-Text Theory (Mel'čuk, 1973); see Mille et al. (2023) for details of levels of representation, and Table 1 for the list of modules that produce these intermediate levels.

A mix of language-independent and language-specific rules build the intermediate representations using additional knowledge contained in language-specific dictionaries. Language-specific rules are needed either because a language phenomenon is highly idiosyncratic in its own nature (e.g. *a* before a vowel in English becomes *an*), or because the conditions of application of a more general phenomenon are idiosyncratic (e.g. the introduction of determiners). From the perspective of multilingualism, there are 3 types (T1-T3) of rules in FORGe: fully language-independent rules (T1, ∼82% of all rules); rules that apply to a subset of languages (T2, ∼6.5 languages on average, ∼3% of rules); and language-specific rules, which apply to one single language (T3, ∼15% of rules).

FORGe uses three types of dictionaries to store:

- Mappings between concepts and lexical units, e.g. *located {GA={lex=lonnaithe_JJ_01}}*.
- Lexical unit descriptions, e.g. *lonnaithe_JJ_01 {lemma = lonnaithe; pos = JJ; preposition_arg2 = i }*, where *i* 'in' is required on the second argument of *lonnaithe*: *lonnaithe i X* 'located in X'.
- Generic language-specific knowledge, such as the type of word order or morphological agreement triggered by surface-oriented dependencies (e.g. a direct object is by default after its governing verb in the sentence, and a determiner receives case, number and gender from its governing noun).

3.2 Extensions for Irish

With respect to dictionaries, we added 457 mappings between concepts and lexical units and as many lexical unit descriptions, and we manually crafted the generic language-specific dictionary. For rules, we implemented 76 rules that apply exclusively to Irish (T3), which represents 2.78% of

[1] Properties such as *municipality*, *district*, or *state* are mapped to the same template as *country*, shown in Figure 1.

[2] We used the publicly available *Translator* module of the *googletrans* (version 3.1.0a0) library.

[3] We refer to a more surface-oriented lexicalisation here, with, e.g., function words, as opposed to the "deep" lexicalisation of the main concepts described in Section 2.

ID	FORGe module	# rl	# T3 GA rl	% T3 GA rl
1	Text planning	553	0	0
2	Lexicalisation	183	0	0
3	Communicative structuring	258	0	0
4	Deep sentence structuring	345	3	0.87
5	Surface sentence structuring	477	17	3.56
6	Syntactic aggregation	215	0	0
7	Referring Expression Generation	237	0	0
8	Word order and agreement resolution	265	17	6.42
9	Morphology processing	201	39	19.4
	All modules	2,734	76	2.78

Table 1: Number of rules, and number and % of Irish-specific (T3) rules (rl) per per FORGe module.

rules; Table 1 shows the breakdown of language-agnostic and language-specific rules per module. We also activated 65 existing T2 rules for Irish.

As Table 1 shows, 4 modules require Irish-specific rules: deep sentence structuring, surface sentence structuring, word order and agreement resolution and morphology processing; next we list the phenomena that required T3 and most T2 rules.

Deep sentence structuring

Relative particles (T3): the particle *a* is introduced to link the modified noun and the main verb in relative clauses; in case of prepositional relatives, the particle has a different form depending on the tense of the verb (present *a*, past *ar*).

Passive (T3): in Irish there are two alternative constructions where a passive form would be used in English. If the data refers to an action/event, an autonomous main verb form is used, e.g. for the triple `Acharya_Institute_of_Technology | established | 2000`, *bunaíodh*, the autonomous form of the verb *bunaigh* 'to establish' is used, as in *Bunaíodh Institiúid Teicneolaíochta Acharya sa bhliain 2000*, 'Acharya Institute of Technology was established in the year 2000'. Alternatively, where a state/location is referred to, e.g. for the triple `MotorSport_Vision|city|Longfield`, we have *tá*, the present tense of the auxiliary verb *bí* 'to be', and the past participle *lonnaithe* 'located', as in *Tá MotorSport Vision lonnaithe i gcathair Longfield*, 'MotorSport Vision is located in Longfield'.

Non-verbal copula (T3): Irish has two copular constructions. The verbal copula is used for changeable properties whereas the non-verbal copula *is* is used for more permanent properties; e.g. for `The_Fellowship_of_the_Ring | author | J._R._R._Tolkien` we have *Is é J.R.R. Tolkien a scríobh The Fellowship of the Ring* where *is* connects the author and the book, and the pronoun *é* agrees with the gender and number of the author.

Surface sentence structuring

Determiners (T3): a definite determiner is only introduced on a noun *N* if *N*'s dependent is not a definite noun or a proper noun.

Dependencies (T2, 22 rules in common with Catalan, Greek, Spanish, French, Italian and Portuguese): surface-oriented dependencies are introduced as, e.g., *subject*, *direct object*, *modifier*, etc.

Word order and agreement resolution

Genitive chains (T3): in a chain of genitive elements, only the last element maintains the genitive case, e.g. in the case of 'the length of the runway of the aerodrome', only the last element 'aerodrome' has genitive case as in *Is é fad rúidbhealach an aeradróim 1,095m*.

Word order class (T3): when an element is established as a member of a class, the class name goes right after the copula, as in *Is milseog é Bionico* 'Bionico is a dessert'.

Possessive pronoun agreement (T3): the semantic number and gender of a possessor triggers agreement on the possessed. In the case of the triple `India | leader | T._S._Thakur`, the copular construction generates the text *Tá T.S. Shakur ina cheannaire ar an India*, 'T. S. Thakur is a leader of India', where we have the present tense of the verbal copula *bí*, followed by the subject *'T. S. Thakur'* and the subject complement *'ina cheannaire ar an India'*. The complement has a possessive pronoun *ina* that agrees in gender and number with the subject, i.e. *ina* is masculine singular reflecting the subject 'T. S. Thakur' and it triggers masculine singular agreement on the noun *cheannaire* 'leader'.

Ellipsis (T3): some rules look for pronouns to elide, in particular in relative and non-verbal copular constructions. In addition, Irish is a VSO language so a specific rule checks for repeated sub-

jects on the right of the verb and replaces them with pronouns.[4]

Order between siblings (T2, 29 rules in common with Catalan, Greek, Spanish, French, Portuguese and sometimes Italian): for instance, in many languages, the determiner usually goes before all other dependents of the noun.

Morphology processing

Concatenations (T3): *don* is a contraction of *do an* 'for the' as in *Scríobh Nicholas Brodszky an ceol don scannán* meaning 'Nicholas Brodszky wrote the music **for the** film'.

Prefixes (T3): vowel-initial masculine nouns following the determiner *an* receive a *t-* prefix as in *Rugadh an* ***t-aisteoir*** *Bill Oddie in Rochdale* meaning 'The actor Bill Oddie was born in Rochdale'. The preposition *le* triggers a prefix *h-* on following nouns starting with a vowel, and some past verbs get the prefix *d'*.

Mutations (T3): word-initial mutations are common in Irish and fulfil many grammatical functions, for example the noun *cathair* 'city' has various mutations depending on the number and gender of the possessive pronoun, e.g. there is lenition in *mo* ***ch****athair* 'my city', eclipsis in *ár* ***gc****athair* 'our city' and no mutation in *a* ***c****athair* 'her city'.

Verbal Adj/N, Prep. declension, V flags (T3): other rules cover the conversion of some adjectives and nouns into their verbal counterparts, the inflection of some prepositions and the insertion of a tag that flags vowel-initial verbs, as required by the morphology generator.

4 Generation of morphological inflections

We describe here the morphology generation (4.1) and its interface with FORGe (4.2).

4.1 Irish NLP Tools

The Irish NLP tools suite[5] includes finite-state transducers for Irish morphology generation (Dhonnchadha et al., 2003). These tools handle tokenisation and morphological analysis/generation of the inflected forms of Irish headwords coded in the finite-state lexicons. The tools were initially developed using xfst (Xerox finite state tools) (Beesley and Karttunen, 2003) and later converted to use foma tools (Hulden, 2009).[6] Finite-state transducers model a two-level morphology where a lexical description is mapped to a surface form, e.g. déan+Verb+VT+FutInd maps to the future tense form *déanfaidh* of the transitive verb *déan* 'make'. The transducers can be used to generate inflected forms of words for NLG and CALL applications, and the same transducers work in the opposite direction for morphological analysis as part of NLP applications including PoS tagging and parsing.

4.2 Interfacing FORGe with Irish NLP tools

In order to match the inputs expected by Irish NLP tools, we process FORGe outputs with regular expressions so as to replace reserved characters, introduce a '+' separator between morphological tags, and insert single line breaks between consecutive words of the same text and double line breaks between consecutive texts.

5 Post-processing

The post-processing consists of regular expressions to revert reserved characters to their original form, true-case and clean the texts, and take care of prefixing, hyphenation, contraction, lenition and eclipsis phenomena triggered by the inflected forms of words; see Appendix A for an example.

6 Discussion and Conclusion

We have presented DCU/TCD-FORGe, a fully rule-based pipeline of four modules for Irish text generation at WebNLG'23; sample inputs and outputs for all modules are provided in Appendix A. The BLEU score provided by the organisers[7] (0.167) is significantly lower than FORGe's scores on English at WebNLG'20 (0.406 (Castro Ferreira et al., 2020)); this is likely because we created our lexicalisations without reference to train and dev Irish texts, i.e. surface similarity is likely to be low. However, the gap in BLEU scores between our system (0.167) and the highest-scoring GPT-based system (0.204) (Lorandi and Belz, 2023) is less than 0.04 points; this compares to a corresponding gap of 0.13 points between FORGe and the best English system at WebNLG'20. We will be able to draw more reliable conclusions when the results of the human evaluation are released.

[4] Strictly speaking, this rule belongs to the REG module but since it has the same conditions of application as ellipsis in other languages, it was left in this module for the time being.

[5] https://www.scss.tcd.ie/~uidhonne/irish.utf8.htm

[6] https://fomafst.github.io/

[7] https://github.com/WebNLG/2023-Challenge/tree/main/evaluation/automatic/scripts

Acknowledgements

Mille's contribution was funded by the European Union under the Marie Skłodowska-Curie grant agreement No 101062572 (M-FleNS).

References

Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Uí Dhonnchadha, Caoilfhionn Nic Pháidín, and Josef Van Genabith. 2003. Design, implementation and evaluation of an inflectional morphology finite state transducer for Irish. *Machine Translation*, 18:173–193.

Mans Hulden. 2009. Foma: a finite-state compiler and library. In *Proceedings of the Demonstrations Session at EACL 2009*, pages 29–32, Athens, Greece. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Michela Lorandi and Anya Belz. 2023. Data-to-text generation for severely under-resourced languages with GPT-3.5: A bit of help needed from Google Translate. In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge*, page tbd, Prague, Czech Republic.

Igor A. Mel'čuk. 1973. Towards a linguistic 'Meaning $\leftrightarrow$ Text' model. *Trends in Soviet theoretical linguistics*, pages 33–57.

Simon Mille, Stamatia Dasiopoulou, and Leo Wanner. 2019. A portable grammar-based nlg system for verbalization of structured data. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1054–1056.

Simon Mille, François Lareau, Anya Belz, and Stamatia Dasiopoulou. 2023. Mod-D2T: A Multi-layer Dataset for Modular Data-to-Text Generation. In *Proceedings of the 16th International Conference on Natural Language Generation*, page tbd, Prague, Czech Republic.

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

A Sample input and output structures

The figures in the next page illustrate the generation process starting from an input triple set that corresponds to the following English text:

> *Agra Airport, operated by Indian Air Force, is located in India. Its ICAO location identifier is VIAG.*

Figure 2 shows a WebNLG'23 input, and Figure 3 shows the output of the lexicalisation module. The FORGe, morphology and post-processing outputs are shown in a one-word-per-line format in Table 2. The output Irish text is the following:

> *Tá Agra Airport, reáchtáilte ag Indian Air Force, lonnaithe ins An India. Tá VIAG in a aitheantóir suímh ICAO.*

```
<entry category="Airport" eid="719" shape="(X (X) (X) (X))" shape_type="sibling" size="3">
  <modifiedtripleset>
    <mtriple>Agra_Airport | location | India</mtriple>
    <mtriple>Agra_Airport | operatingOrganisation | Indian_Air_Force</mtriple>
    <mtriple>Agra_Airport | icaoLocationIdentifier | &quot;VIAG&quot;</mtriple>
  </modifiedtripleset>
</entry>
```

Figure 2: A sample WebNLG input with 3 triples

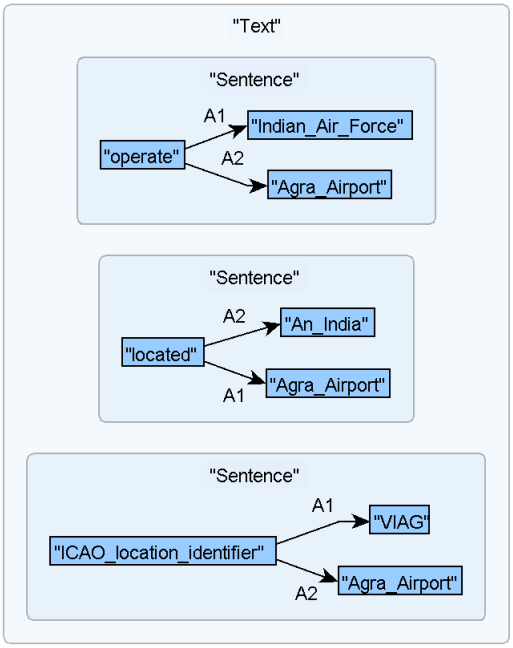

Figure 3: Lexicalisation output: instantiated PredArg templates

FORGe	Morphology	Post-processing
bí+Verb+PresInd	tá	Tá
Agra_Airport+Noun+Masc+Com+Sg	Agra_Airport	Agra Airport
,	,	,
reáchtáilte	reáchtáilte	reáchtáilte
ag	ag	ag
Indian_Air_Force+Noun+Masc+Com+Sg	Indian_Air_Force	Indian Air Force
,	,	,
lonnaithe+Adj+Masc+Com+Sg	lonnaithe	lonnaithe
i	i	ins
An_India+Noun+Masc+Com+Sg	An_India	An India
.	.	.
bí+Verb+PresInd	tá	Tá
VIAG+Noun+Masc+Com+Sg	VIAG	VIAG
i	i	in
a	a	a
aitheantóir+Noun+Masc+Com+Sg	aitheantóir	aitheantóir
suímh	suímh	suímh
ICAO+Noun+Masc+Com+Sg	ICAO	ICAO
.		

Table 2: FORGe, morphology and post-processing outputs (one word per line for convenience)