AMERICAN UNIVERSITY OF BEIRUT

MODELS AND RESOURCES FOR ARABIC DATA-TO-TEXT GENERATION

by
ROUDY SAMIR TOUMA

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
Jan 2022

AMERICAN UNIVERSITY OF BEIRUT

MODELS AND RESOURCES FOR ARABIC DATA-TO-TEXT GENERATION

by

ROUDY SAMIR TOUMA

Approved by:

Dr. Hazem Hajj, Associate Professor — Advisor
Electrical and Computer Engineering — Hazem

Dr. Mazen Saghir, Associate Professor — Member of Committee
Electrical and Computer Engineering

Dr. Wassim El Hajj, Associate Professor — Member of Committee
Computer Science

Date of thesis defense: Jan 15, 2022

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:	Touma	Roudy	Samir
	Last	First	Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

□ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

02-02-2022

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I would like to say a special thank you to my supervisor Dr. Hazem Hajj for his support and guidance throughout my graduate studies. I would also like thank the committee for their valuable inputs during the thesis defense and my friends for being a great support system. Last but not least, I would like to thank Ms. Doja for all the times she helped me improve in spirits.

An Abstract of the Thesis of

Roudy Touma for Master of Engineering
Major: Electrical and Computer Engineering

Title: Models and Resources for Arabic Data-to-text Generation

Resource Description Framework (RDF) is the standard for representing structured knowledge on the Web. It is based on entities such as facts, events, and the relationships between them. RDF verbalizers are important to generate good quality textual descriptions from such RDF data. Despite the significant work done for the English language, no efforts have been directed towards low-resource languages like the Arabic language. This work promotes the development of RDF data-to-text (D2T) generation systems for the Arabic language by introducing a new Arabic dataset (AraWebNLG). A comparative study between multiple sequence-to-sequence models is also presented while studying the transfer of knowledge from pre-trained language models (AraBERT, AraGPT2 and mT5) to overcome data limitations. The analysis involves numerical metrics (BLEU and Perplexity scores) as well as task-specific metrics related to the accuracy of the content selection and fluency of the generated text. The results highlight the importance of pre-training on large corpus of Arabic data as the AraBERT initialized model is the best performing among the others. Text-to-text pre-training using mT5 is also able to achieve competitive results even with multilingual weights.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Resource description framework (RDF) data is the standard for representing structured knowledge on the Web. It is based on entities, representing people, places, and abstract concepts, such as events. They can also contain facts and can be used to form meaningful relationships between them. RDF Data-to-text (D2T) generation is the task of converting this structured non-linguistic data into natural language. Unlike other natural language generation (NLG) systems such as, conversational chatbots and question answering where the input is generally in the form of unstructured text, the input in D2T tasks contains structured information such as tables and knowledge graphs. D2T generation can be used in several commercial applications including voice assistants and question answering bots. Also, media organizations like The New York Times and BBC rely on RDF data as one of the main methods of publishing. An example of D2T generation is illustrated in Figure 1.1 where a knowledge-graph describing the relationship between sports teams is transformed into a meaningful and factual sentence.

Previous work in this area can be broadly categorized as either pipeline or end-to-end approaches. In pipeline-based approaches, the input data is converted into natural language through several intermediate transformations. However, the recent research is shifting towards adopting end-to-end approaches where the input data is directly mapped to an output text using encoder-decoder architectures. End-to-end methods have shown efficacy for short sentences when compared to pipeline-based methods [1] [2] [3]. The main challenges when developing end-to-end models are ensuring proper content selection, data fidelity and text fluency at the same time. Recently, there has been an increased interest in pre-trained language models like T5 [4] and BART [5] for D2T generation. Language models are trained on large corpus of text which makes them excel at generating natural-sounding text. These language capabilities allow them to address the challenge of text fluency while maintaining proper content selection and understanding.

Despite the significant work done for D2T generation for English, few efforts have being directed towards low-resource languages like Arabic. In this work, we address the existing challenges related to D2T, which are previously addressed

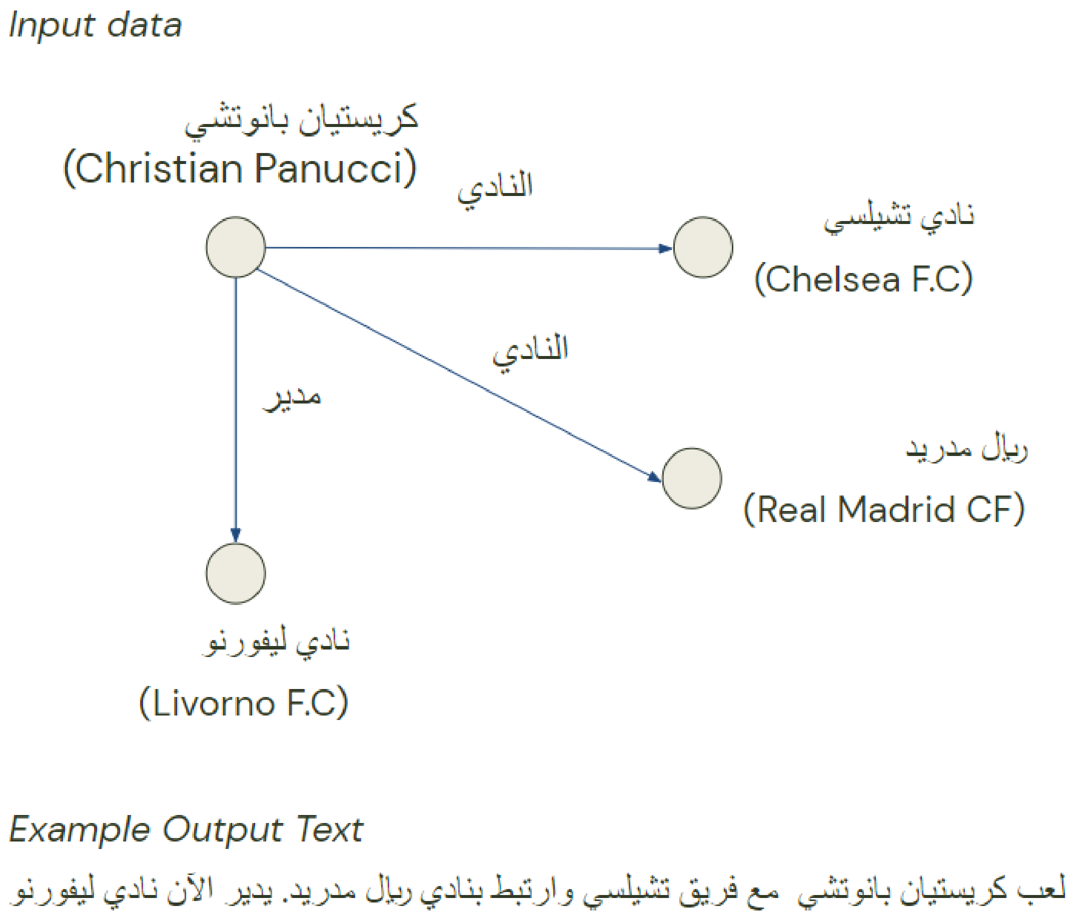


Figure 1.1: Example of D2T generation from the knowledge graph relating sports teams

in English, for Arabic while focusing on additional challenges of learning with low-resource availability and generalizing to unseen domains. Therefore, the contribution of this work is as follows:

1. We propose the first D2T news generation in the Arabic language with comparable results to the English language.

2. We introduce AraWebNLG, the first Arabic dataset for the task of D2T generation based on the English dataset WebNLG [6].

3. We provide a comparative study between multiple end-to-end models including pre-trained language models to address the challenges of low-resource availability and overcome data limitations.

The rest of this thesis is organized as follows: Chapter 2 reviews the literature on D2T generation in English as well as the NLG progress in Arabic.

The AraWebNLG dataset is introduced in Chapter 3 along with our proposed methodology. Chapter 4 details the performed experiments and presents a comprehensive analysis of the results. Finally, concluding remarks and directions to future work are provided in Chapter 5.

Chapter 2

Literature Review

2.1 Background into text generation and language modeling for the English Language

Research in text generation has been greatly enhanced over the past years with the advancements of deep learning and neural language models. In this part, we give a brief overview of the general text generation methods for the English language.

Early approaches to automatic text generation mainly relied on rule-based techniques. To generate text, these methods use the semantics and discreet linguistics rules. Even though they can cover a small part of the human language needed for specific applications, these methods are not scalable and cannot be relied on for practical applications because it is very hard to discretize all the rules that cover linguistics. This is why, researchers started shifting towards statistical approaches which formulate language generation as a maximum likelihood estimation problem. This idea is used in the traditional n-gram model. Given a sequence of $n-1$ words, the model can estimate the probability of the following nth utterance. The main limitation of this model is that it cannot be used for large-scale applications which require lengthy texts like news articles. The model can only look at the previous $n-1$ words which limits its ability to capture context and its generalization capabilities.

With the advancements of deep neural networks, language models are becoming more and more advanced and able to capture context over long sequences of text. They can learn the vector representations of text, encode the context and semantics and use these representations to reason and decode proper sequences.

Reccurrent Neural Networks and Sequence-to-Sequence Models (RNNs) can be considered as the classic approach to text modeling and generation. Due to their sequence structure, they are able to process natural language sequential text. Several variations of the RNN models can be found such as the long short-term memory (LSTM) and the gated recurrent unit (GRU). These generally address some of the limitations of the traditional RNN model. In the context of text generation, these systems usually fall under sequence-to-sequence (seq2seq) generation models. Seq2seq models were first introduced by Sutskever et al. [7]. They consist of an encoder network that transforms a series of text inputs into an embedding representation. Next, a decoder network takes this vector representation and predicts the output sequence. Seq2seq models have been widely used as standard approaches for natural language generation problems such as machine translation[8], text summarizing[9] and story generation[10].

Conventional Seq2seq models have several limitations. They are computationally demanding and have trouble maintaining long-term dependencies in text sequences. In addition to that, input text is usually compressed into a fixed vector space where all words have the same weight. This limits the ability of Seq2seq models to capture context and extract the main important information from text. To address this issue, Google introduced in 2018 a seminal work "Attention Is All You Need" [11]. In short, attention mechanisms assigns different weights to the input tokens based on how relevant they are to the context. Instead of processing each token by itself, attention architectures create a mapping between the inputs to learn their relationship and dependencies. This marked the start of attention-based transformer architectures in natural language modeling application as a replacement for recurrent neural networks.

BERT [12], Bidirectional Encoder Representations from Transformers, is a language model based on the transformer encoder architecture. It was released by Google in 2018 and since then have become one of the most important language models for NLP. What makes BERT remarkable is its ability to capture context in sentences due to its bidirectional property. Unlike previous word embedding models like *Word2Vec* [13], *FastText* [14], and *GloVe* [15] which have one single encoding for each vocab regardless of how it is used in the context of the sentence, BERT on the other hand generates embedding representations for the words based on their specific context in the input. BERT has achieved excellent results on discriminative tasks like text classification and sentiment analysis. However, when it comes to text generation, it lacks due to its bidirectional property. The model is trained on the masked token prediction and take into account both the future and past context of sentences. When it comes to text generation, the model has no access to the future context since it is trying to generate them. An attempt to use the knowledge of BERT-like models for text generation can be found in Distill-BERT [16]. The authors use knowledge distillation from a "teacher" BERT model and transfer this knowledge to "student" Seq2Seq model during training in a way that the Seq2Seq model generates new text while relying on BERT's contextual knowledge.

Transformer-XL [17], is also based on the transformer architecture with a recurrence mechanism built on top of it. This mechanism allows long term dependencies between the text. It also introduces a new positional encoding which is based on relative distances instead of absolute encodings. This model is used in

XL-Net [18], an autoregressive language model which learns bidirectional contexts over all permutations of the input text. The model overcomes the limitations of BERT and has been widely used for generative tasks by researchers in the field.

GPT-2 [19], Generative Pretrained Transformer 2, is another auto-regressive transformer based model widely used for text generation tasks. The model relies on a decoder-only architecture and is pre-trained on the task of predicting the next token at every time step given the previous tokens. GPT-2 was introduced by OpenAI in 2018 and has been used since then for various generative tasks like conversational dialogues [20], story [21] and poetry [22] generation. Also, most recently, GPT-3 [23] was introduced. The model is an enhanced version of GPT-2 with a very complex architecture (175 billion parameters). The model can be used as a few-shot learner where very few data is needed to use the model on downstream task. Due to ethical concerns regarding the use of GPT-3 for malicious intent, OpenAI decided to only provide the model to selected few researchers.

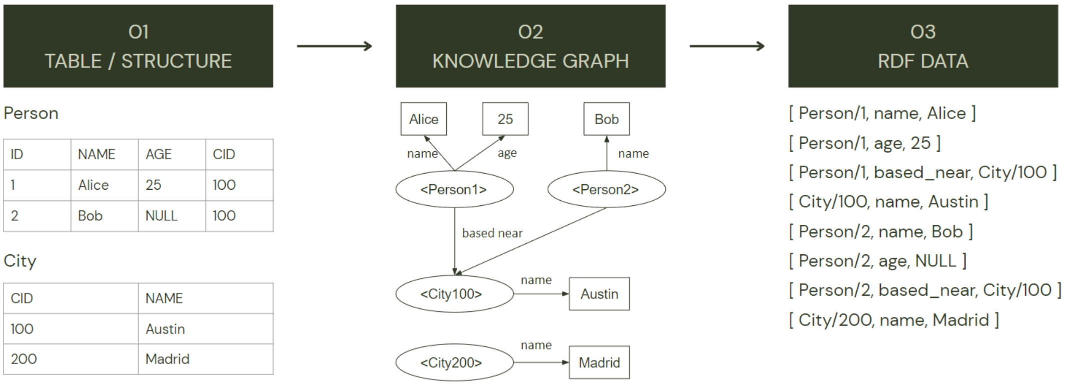
Recently, [24] introduced T5, a text-to-text-transfer-transformer model. It is based on the encoder-decoder transformer architecture and masked language modeling. The authors reframe all natural language processing (NLP) tasks into a unified text-to-text format. The model is pre-trained on a very large corpus of data and has achieved state-of-the-art (SOTA) results on multiple downstream tasks including question answering, summarization and more. This model is also available as mt5 with multilingual pre-trained checkpoints [25].

To skip the costly pre-training of text-to-text models, [26] have shown the importance of using pre-trained checkpoints for sequence generation tasks. The authors have experimented with warm-starting the transformed-based encoder-decoder model with checkpoints from pretrained language models including BERT [12] and GPT-2 [19]. The results show that warm-starting encoder-decoder models with pre-trained weights are able to achieve competitive results in sequence generation tasks when compared to large pre-trained language models such as T5 at a fraction of the training cost.

Another research area that is gaining popularity is the conditional text generation. The previously mentioned language models can generate remarkably realistic text. However, even though attention mechanisms help guide the context of the text generation, the models can diverge and generate text that is not controllable. To help deal with this issue, conditional generation tools are introduced. Instead of relying only on the context and sequence of textual inputs to generate text, external inputs are used to guide the generation process. For example, researchers have used arithmetic transformations to control the text generation output of the encoder [27] [28] and generate text that conforms to a certain textual style. variational autoencoders (VAE) are also used to guide the generation process of the decoder by stochastically sampling from a certain latent distribution. This method is used in [29] to generate texts conforming to certain topics. Furthermore, Generative Adversarial Networks (GANs) are used to provide external feedback to control the the external input's effect on the generator. A discriminator is generally used to guide the sampling of the generator. This can be seen in [30] and [31] where reinforcement-learning reward signals coming from the discriminator are used to improve the quality of the text generation and encourage the sampling of creative, non-repetitive tokens.

2.2 Data-to-text generation (D2T) Generation in English

First, we explain how can data be transformed into natural language. One way of doing so would be through an intermediate representation called resource description framework or RDF. As can be shown in Figure 2.1, the tabular data explaining details about people and cities can be transformed into a form of knowledge graph where the different entities of the table are related together. Them, this knowledge graph can be used to extract triplets connecting all the nodes together. These triples are called RDF triples. This way, the problem of data-to-text generation can be transformed into rdf-to-text generation.



Person

ID	NAME	AGE	CID
1	Alice	25	100
2	Bob	NULL	100

City

CID	NAME
100	Austin
200	Madrid

[Person/1, name, Alice]
[Person/1, age, 25]
[Person/1, based_near, City/100]
[City/100, name, Austin]
[Person/2, name, Bob]
[Person/2, age, NULL]
[Person/2, based_near, City/100]
[City/200, name, Madrid]

Figure 2.1: From Data to Natural Language - Intermediate representation through RDF triples

D2T generation systems can be separated based on the end-result they generate. Some previous work focus on generating sentences while others are more concerned with generating documents. When it comes to commercial journalism tools, these focus on generating sentences describing the data. This way, the journalist would be responsible for combining sentences together, choosing which info is important and then adding their own analysis. As for the research work, the work done on sentence generation can be thought of as the building block for the work done on document generation since document generation is a more challenging task for data-to-text generation models. As for document generation, these are usually restricted to purely reporting articles in commercial applications where no insights or additional details are needed. Also, in document generation, there is more room for error. This restricts the generation process in commercial applications to a fully controlled template that guarantees that only the input data is written. As for research work, people work on document generation because it provides a more challenging task for data-to-text models. And can evaluate the performance on a content selection and content planning more clearly because this is more noticeable in document rather than sentences. One thing to consider in terms of datasets, is that document generation requires datasets that are fact grounded meaning that the paragraphs only include details about the input data. This is hard to do because most news articles include external analysis that is not found in the input data which tends to confuse the models. For example, game reports do not closely follow the game statistics, but in practice they contain a background knowledge, interpretation, insight into the game, and quotes that are not present in the official statistics. This makes it difficult to find noise free datasets that are fact grounded with the input data. In this work, we mainly focus on sentence-level D2T generation system.

The conventional approach for solving the D2T generation problem relies on pipeline systems. In this approach, linguistic realization is separated from content selection and planning. Three main modules can be generally found in pipeline systems. First, there's the text planner that is responsible for selecting what to say from the input information. It is a macro-planner which selects the content and decides on the high level structure of the text. Second, there's the sentence planner which is responsible for how to say the content. It is a micro-planner responsible for lexicalization, sentence aggregation and referring expression generation. Finally, the linguistic realizer which is responsible for writing the text based on syntactic and morphological rules. This architecture can be shown in Figure 2.2.

The intermediate steps can be template-based which rely on specific sets of handcrafted rules [1] [2] [32]. However, one challenge with templates is that the text can be too structured and lack human flow. Researchers have tried to solve this by relying on ranking models [32] [33] and paraphrasing models [34]. Another challenge is that templates are hard to construct from scratch. People have tackled this by working on automatic template generation systems [35]. The main advantage of templates is having full control over the output with a guarantee that no misinformation or grammatical mistakes would be present in the output. Pipeline architectures can also include neural networks such as GRU and transformer models [3] [33] as intermediate steps. The main challenge here is the concatenated error from the different modules. Also, by discretizing the generation process into defined steps, a generation gap is created. A writer usually has a high-level plan of the text they are going to write, but they would usually change things and adjust sentences as they write. In pipeline methods, this is usually not the case because the strategic and tactical components of the writing process are separated. So for example, a generation system might

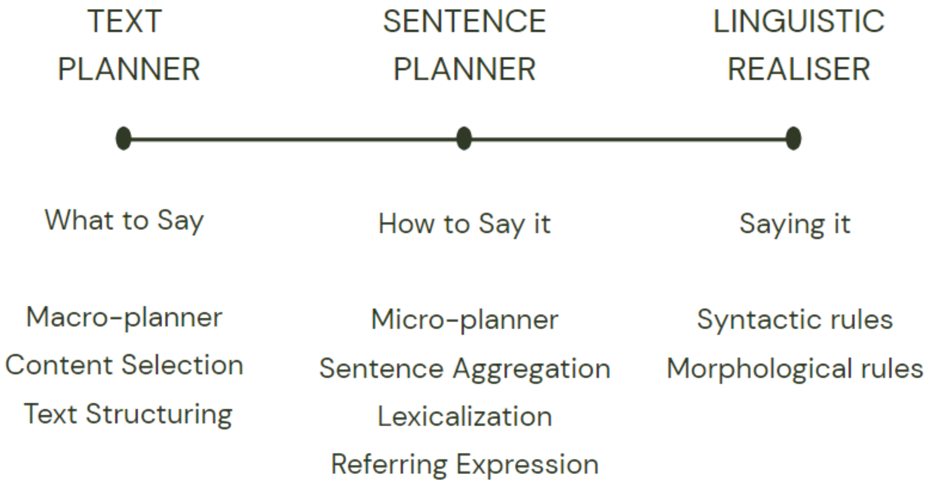


Figure 2.2: Standard architecture of pipeline-based data-to-text generation system including three main components

determine a particular sentence ordering during the sentence planning stage, but this might turn out to be ambiguous once sentences have actually been realized.

Recently, end-to-end approaches have proven to be a good alternative to the labor-intensive and domain-specific pipeline methods. These work by directly mapping the input data to natural text using encoder-decoder architectures as can be seen in Figure 2.3. In end-to-end approaches, the distinction between content selection, content planning, and linguistic realization is blurred out as everything is done simultaneously. End-to-end encoder-decoder models relying on GRU, LSTM and Transformer models are used in the literature [1] [2] [3] to address the D2T problem. Because everything is done simultaneously, one challenge for end-to-end models is the content selection and planning especially when it comes to generating lengthy text. In D2T generation, it is important to generate factual output that matches the given input. Because neural networks can lack interpretability, they have the tendency to diverge from the given input or miss out on important information especially when the input data becomes longer. To address this challenge, researchers have used separate content selection modules without sacrificing the end-to-end training [36] as well as hierarchical models [37]. Another challenge is the text fluency and generalization to other domains. When trained on a specific set of data, these models struggle when generalizing to unseen domains. The research focus in this area is to rely on pre-trained language models to benefit from the language capabilities of these models to generate fluent text while maintaining good content selection.

[4] and [5] have investigated the fine-tuning of a pre-trained T5 model for D2T generation tasks. The model achieved SOTA results on multiple D2T datasets including WebNLG [6], MultiWoz [38] and ToTTo [39] benchmarks.

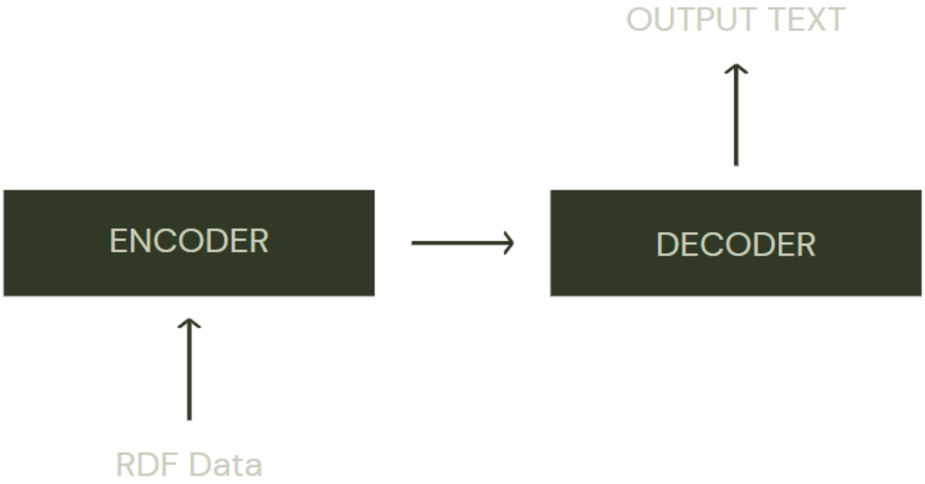


Figure 2.3: End-to-end encoder decoder that directly transcribes RDF triples into natural language

Automated journalism is the generation of news reports from structured non-textual data inputs. Data like sport games results, finance and other statistical domains can be transformed into documented reports by relying on automated data-to-text generation systems. In a paper proposed by Kanerva et al. [40], the authors generate news reports for Finnish ice hockey games. They rely on a system to extract the main events and results of the game and decode this information using an attention-based decoder. The generated text, however, includes false information due to the mismatch in player names and winning teams. Also, the text generated is somewhat discontinuous and requires human editing for viable publications. Another work that tackles a similar problem for soccer articles automation [41]. The authors rely on knowledge graphs build upon content determination, aggregation and lexicalization based on pre-defined templates before generating the resulting text. The main limitation of this approach is that it relies on an explicit knowledge graph that is domain specific which limits its generalizabilty to other domains. Also, defining this knowledge graph can be a tedious task which requires a lot of fine-tuning. Hence, the main research challenge in this area is the structure control of the output article and maintaining a coherent arrangement of the extracted information.

2.3 NLG in Arabic

The abundance of huge datasets and large-scale computing power for the English Language has accelerated the progress of the research in the area of text generation by introducing big language models like Grover and GPT. However, when it comes to low resource languages like Arabic, these datasets are not available and are very hard to collect and label. Also, generalizing the English models to the Arabic language would require machine translation systems which add an extra layer of error. In addition to that, languages like Arabic have specific intrinsic properties and challenges that cannot be modelled and represented through English data. Unfortunately, this has relatively stalled the progress of the research in Arabic language generation as the most important research areas in English language generation have not been extensively explored for the Arabic language. Most previous work falls under Natural Language Understanding (NLU) and is still restricted to few tasks like text classification and sentiment analysis. There has been, however, some previous efforts in researching Arabic language generation areas. Arabic Machine Translation (AMT), for example, has been receiving a good amount of attention by researchers as detailed in a recent survey by [42]. Also, a recent survey [43] examined the available chatbots for Arabic. All of these, however, rely on non-learning pattern matching techniques to sample dialogue responses from pre-defined answers. To a lesser degree, researchers have addressed image captioning [44] and poetry generation [45] by relying on small datasets and simple recurrent neural network based models. Finally, an abstractive Arabic text summarizer is presented in [46]. In the context of news articles, automatic summarizing techniques are used to generate news headlines. This problem was addressed in [47], where the authors used statistical and extractive techniques to generate headlines.

Despite the big research progress in the area of D2T generation in the English language, there hasn't been any attempt on tackling this research problem for low-resource langauges like Arabic. This is mainly related to the lack of Arabic datasets and resources.

Recently, the research on Arabic language processing and generation has focused on introducing pre-trained Arabic language models such as hULMonA [48] and AraBERT [49] which have proved to be very important for NLP tasks. AraBERT is an Arabic pretrained version of Google's BERT [12] that is trained on large Arabic corpora extracted from Arabic news sources. The model achieves state-of-the-art results on multiple downstream tasks including Sentiment Analysis, Named Entity Recognition, and Question Answering. Most recently, [50] have introduced AraGPT2, the first advanced Arabic language generation model. The model is based on the GPT2 architecture [19] which consists of a stack of transformer decoder blocks followed by a dense layer. It relies on the auto-regressive language modeling objective and it is trained on a huge corpora of Arabic text. Specifically, the model is capable of producing good quality Arabic text and achieves successful results on different tasks including Synthetic News Generation, and Question Answering. Because they are trained on a huge Arabic corpus, AraBERT and AraGPT2 are able to accumulate a great lexical and pragmatic knowledge which makes them good candidates for the task of D2T generation which requires both good language understanding and generation.

In this work, we address the D2T-related challenges for the Arabic language. The challenges, previously addressed in the English language, include content selection & planning, factual output generation and text fluency. Another main challenge we are targeting is the low-resource availability for Arabic. To do so, we introduce a new Arabic dataset. We also leverage the pre-trained weights from Arabic language models including AraBERT [49] and AraGPT2 [50] and we compare them against the multilingual text-to-text model mT5 [25].

Chapter 3

Proposed Method

3.1 Dataset: AraWebNLG

Since no dataset is available for D2T in the Arabic language, we translate WebNLG[6], an existing and widely used dataset in the English language. The dataset contains 21,855 data/text pairs where the text is a verbalisation of the data. Also, a Russian version of the WebNLG dataset [51] was recently introduced. It contains around 15k instances of data/text pairs translated from the English dataset. The WebNLG dataset contains multiple categories including Artist, Food, Celestial-Body, SportsTeam, Politician, University, etc. The dataset was originally created to encourage the development of verbalisers that can generate short text from RDF data, i.e., a knowledge graph. For instance, the knowledge graph shown in Figure 1.1, can be transformed into 3 RDF triples relating all the different nodes as such:

1. نادي ليفورنو | مدير | كريستيان بانوتشي

   A.S. Livorno Calcio | manager | Christian Panucci

2. كريستيان بانوتشي | النادي | نادي تشيلسي

   Christian Panucci | club | Chelsea F.C.

3. كريستيان بانوتشي | النادي | ريال مدريد

   Christian Panucci | club | Real Madrid C.F.

A total of 7,001 distinct data/text pairs were selected from the dataset and translated to Arabic using Google Translate's API[1]. Next, in-house crowd-sourcing is used to post-edit the translations and manually correct the wrong translations. Finally, quality checking is performed on a small subset of the dataset by human labelers to make sure that the translated samples are reasonable and understandable.

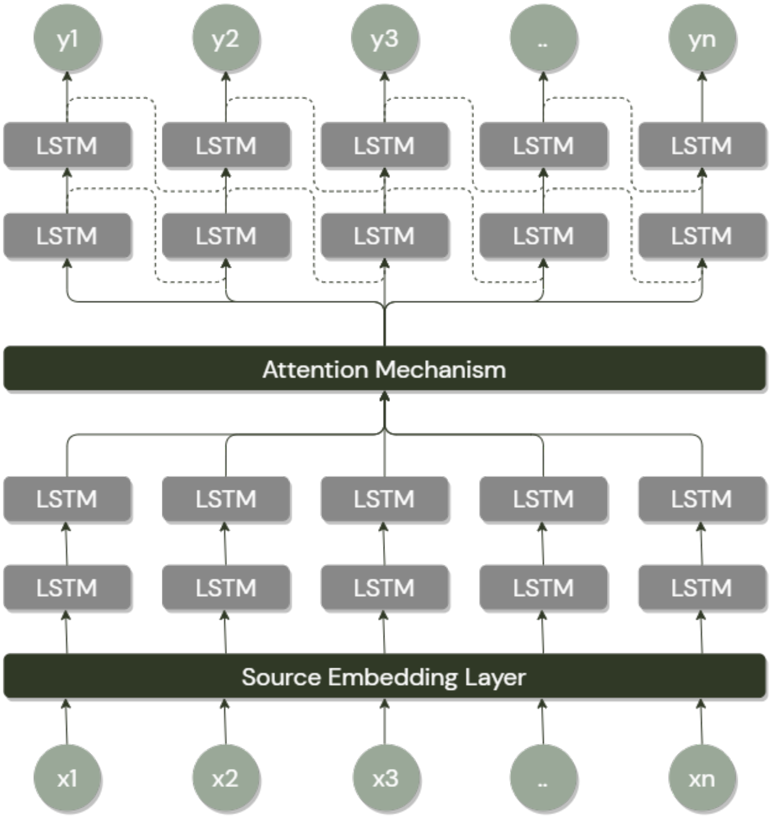


Figure 3.1: Architecture of baseline model

3.2 Baseline model

Similarly to the work done by [3], we rely on an End-to-end sequence-to-sequence (Seq2Seq) model as a benchmark model. The encoder consists of an embedding layer (of size 100) and 2 LSTM layers. As for the decoder, stacked LSTM layers are used with dropout to avoid overfitting. A global attention mechanism is also used with the Seq2Seq model. The architecture of the model is shown in Figure 3.1.

[1] https://pypi.org/project/googletrans/

Example Text
ولد عبد السلام علي أبو بكر في ولاية
النيجر وكان رئيس موظفي الدفاع في نيجيريا
Farasa Segmentation
ولد عبد ال +سلام علي أبو بكر في ولاي +ة
ال +نيجر و +كان رئيس موظف +ي ال +دفاع
في نيجيريا
BPE Tokenization
ولد عبد ال +سلام علي أبو بكر
في ولاي +ة ال +نيج +ر وكا +ن
رئيس موظف +ي ال +دف +اع في نيجيريا

Table 3.1: Example of Arabic Tokenization

Because the baseline model is not pre-trained on any Arabic corpus, we need to reduce the size of the vocabulary so that it is able to learn proper language relationships. To reduce lexical sparsity, the samples in the dataset are segmented using Farasa Segmenter [52]. It is an SVM rank-based segmenter used for Arabic text tokenization. It uses a variety of features and lexicons to rank different possible segmentations of a word. These features include the likelihood of stems, prefixes, suffixes, and their combination; presence in lexicons containing valid stems and named entities; and underlying stem templates. An example of a sample before and after segmentation can be seen in Table 3.1.

3.3 Warm-starting Encoder-Decoder Transformer

Our proposed model for D2T generation is based on the transformer encoder-decoder architecture [11]. The encoder is a stack of encoder blocks and the decoder is a stack of decoder blocks, followed by a dense layer, called language model head. The architecture can be shown in Figure 3.2. Because we are dealing with a limited-sized dataset, we want to leverage transfer of knowledge from pre-trained language models in order to enhance performance, improve generalizability and fluency. Similarly to the work done by [26], we study two possibilities to warm-start an encoder-decoder model:

1. **BERT2BERT**: where we initialize both the encoder and decoder part from an encoder-only model checkpoint (AraBERT).

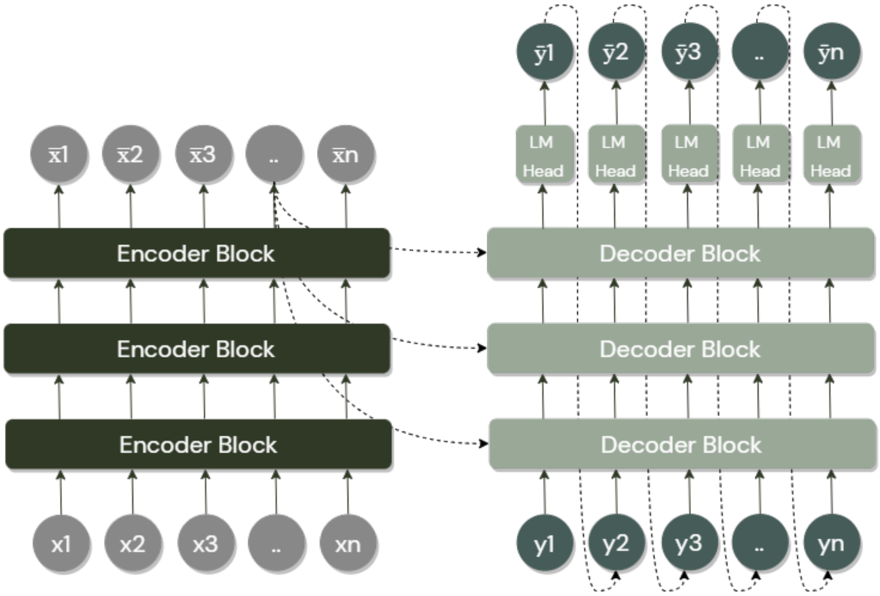


Figure 3.2: Architecture of the proposed encoder-decoder transformer initialized with AraBERT and AraGPT2 checkpoints.

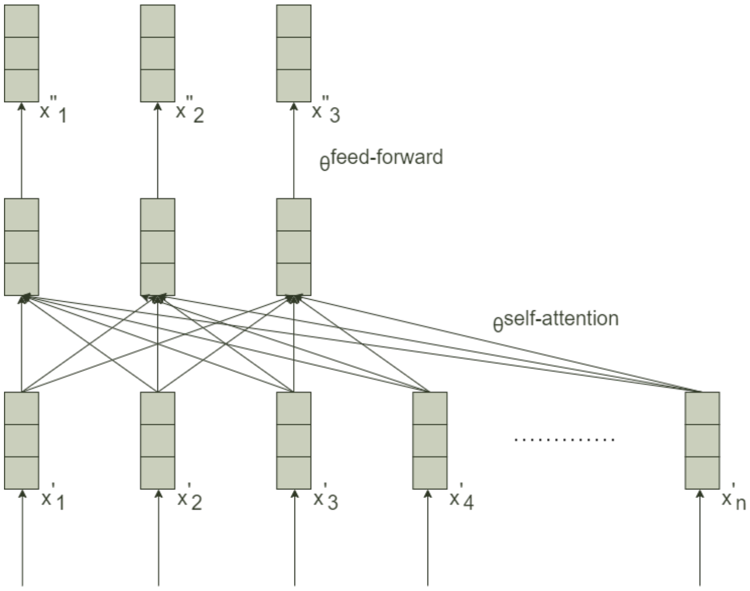


Figure 3.3: Architecture of encoder block which is the same as BERT's encoder block.

2. **BERT2GPT**: where we initialize the encoder part from an encoder-only model checkpoint (AraBERT), and the decoder part from and a decoder-only model (AraGPT2). The intuition here is that we can benefit from BERT's great input understanding and GPT-2's great text generation at

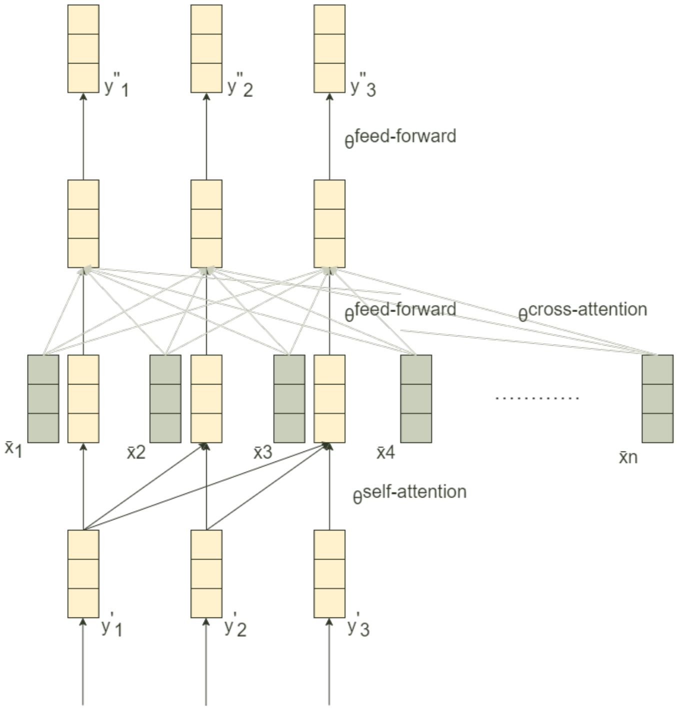


Figure 3.4: Architecture of decoder block.

the same time.

In order to initialize an encoder-decoder with pre-trained BERT or GPT-2 weights, the architectures are compared and common layers are initialized with the pre-trained weight parameters of the respective layers. The rest of the layers are randomly initialized. The encoder block, shown in Figure 3.3, which consists of one bi-directional self-attention layer and two feed-forward layers, matches perfectly with the architecture of BERT. This means that the layers of all the encoder blocks are initialized with the weights of the BERT encoder blocks. As for the decoder block, shown in Figure 3.4, it consists of one unidirectional self-attention layer, one cross-attention layer, and two feed-forward layers. So, in the case of a BERT-initialized decoder, the cross-attention layers are randomly initialized, the feed-forward layers are initialized using BERT's weights, and the unidirectional self-attention layers are initialized using the bi-directional BERT layers. Finally, the language model head layer on top of the last decoder block is initialized using BERT's word embedding layer. In the case of a GPT2-intialized decoder, the architectures are more similar. The decoder block of GPT-2 can be shown in Figure 3.5. Only the cross-attention layers are randomly initialized while the rest of the layers are equivalent to GPT2's layers including the language model head, uni-directional self-attention and feed-forward layers.

Concerning the arabic tokenization used with these models, the Byte-Pair-Encoding (BPE) tokenization is used. This scheme is typical with transformer

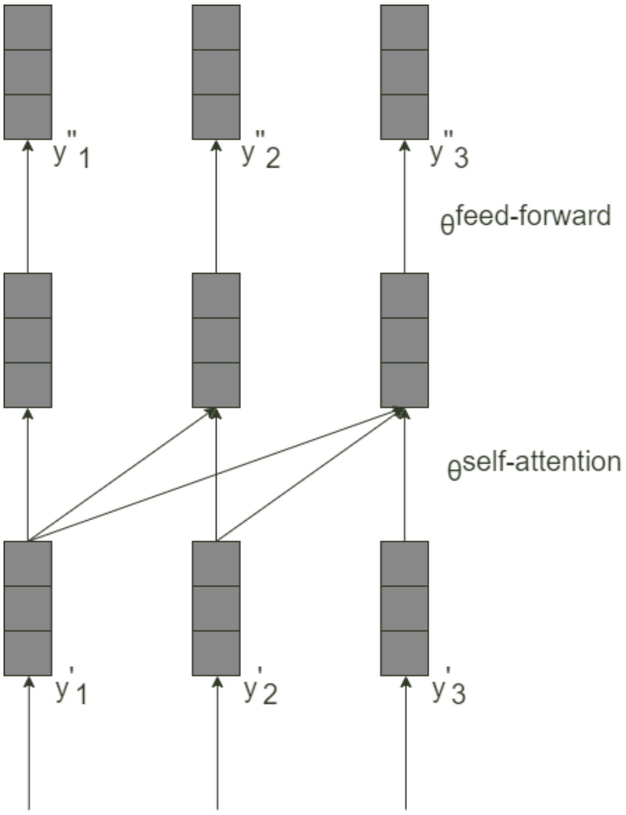


Figure 3.5: Architecture of decoder block of GPT-2.

models and is able to build words it has never seen before by using multiple subword tokens, and thus requires smaller vocabularies. An example of a tokenized sentence can be seen in Figure 3.1.

3.4 T5

T5 is a type of Transformer that is based on the encoder-decoder transformer architecture presented by [11] with some exceptions. T5 was introduced by [24] with the goal of creating a unified architecture that can learn multiple language tasks at once. In this work, we fine-tune the model on the task of D2T generation. Since a pre-trained Arabic version of this model is not yet available, we rely on the multilingual version for fine-tuning. One main difference between the T5 model and the previously mentioned BERT2BERT and BERT2GPT models is that T5 is that T5 is pre-trained on text-to-text generation meaning that the encoder and decoder were pre-trained together on a large corpus of data. While in the case of the other models, the encoder and decoder are both pre-trained separately. Because the model can be used for multiple tasks at once, it expects a task-related prefix. In this case, we append the prefix "data2text: " at the beginning of all the samples.

Chapter 4

Results

4.1 Experimental Setup

The transformer-based models including BERT2BERT, BERT2GPT and T5 were developed using the Huggingface transformers library [1]. The baseline model is developed using OpenNMT library [53]. All models are trained using the same dataset partitioning (80% training, 10% validation, 10% test) which is the same split used in the original WebNLG data.

4.1.1 Quantitative Evaluation

The numerical performance evaluations of the models are summarized in Table 4.1. Here, we rely on the BLEU (bilingual evaluation understudy) and perplexity (PPL) scores. As can be seen from the table, the BERT2BERT model consistently outperforms the other models since it has the lowest PPL and highest BLEU scores. This validates the importance of initializing the BERTB2ERT model with pre-trained AraBERT checkpoints. The results also show that mT5 is the second best performing model with a 4.89 reduction in BLEU score and an increase of 0.37 in PPL score from BERT2BERT. However, the intuition that using GPT-2 model as a decoder for better text generation alongside BERT is not completely validated since it exhibites 5.7 points reduction in BLEU score and 0.867 increase in PPL score from BERT2BERT. The baseline model has the lowest performance with high perplexity score of 8.025 and a low BLEU score of 16.72 which indicates the content can be hard to understand. The numerical results highlight the importance of pre-training on larger corpus of text to address the challenge of limited resource data as all the pre-trained models exhibit better performance than the baseline.

[1] https://github.com/huggingface/transformers

Model	**BLEU**	**PPL**
Baseline	16.72	8.025
BERT2BERT	**25.65**	**1.383**
BERT2GPT	19.95	2.25
mT5	21.76	1.60

Table 4.1: Performance of the models on the test set

4.1.2 Qualitative Evaluation

Numerical metrics such as the BLEU and PPL scores are not completely sufficient to evaluate the models' performance on the D2T generation task. Therefore, we conduct qualitative evaluation where human raters are presented with the input data in triple form along with the predicted texts from each model, i.e., a total of 4 sentences per example. The raters are asked to judge the prediction along two axes on a scale from 0 to 5 where 0 conveys terrible performance and 5 for excellent performance. The first metric is related to the accuracy or faithfulness to the input data. This metric is used to gauge whether the output texts convey the same information as the input information and whether there are missing and/or wrong information included in the output. The second evaluation metric is the fluency of the text. This is an aggregate score related to the linguistic realization which accounts for whether the text flows in a natural, easy to read manner and whether wrong expressions and/or repetitions are present in the text.

In order to asses the agreement between the raters, Fleiss' kappa (κ) is used. This is a statistical measure that calculates the inter-observer agreement taking into account the expected agreement by chance as shown in Equation 4.1 where P_o is the observed agreement and P_e is the expected agreement.

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{4.1}$$

As shown in Table 4.2, BERT2BERT substantially outperforms all other models on the faithfulness metrics with a score of 4.044 while mT5 stands in second place at 3.467. BERT2GPT and the Seq2Seq baseline model perform poorly with low scores of 2.782 and 2.290, respectively. As for the fluency scores, they are relatively decent for all models. However BERT2BERT is still better performing with a fluency score of 4.264 followed by mT5 with 3.894 score. The statistics on the raters' agreement shown in Table 4.3 indicate a Fleiss kappa score of 0.383 for the fluency metric which reflects a fair agreement between the raters. A value of 0.52 for the Fleiss kappa score on the faithfulness metric indicates a moderate agreement.

Model	Faithfulness	Fluency
Baseline	2.290	3.484
BERT2BERT	**4.044**	**4.264**
BERT2GPT	2.782	3.691
mT5	3.467	3.894

Table 4.2: Human evaluation metrics on the test data

	Faithfulness	Fluency
Fleiss Kappa	0.520	0.383

Table 4.3: Statistics on rater's agreement: Fleiss' Kappa

4.1.3 Qualitative Assessment

As can be seen from the numerical and human evaluations, BERT2BERT provides the best results with the highest score of 4.044 on the faithfulness metric. This reflects BERT's abilities when it comes to data understanding as a state-of-the-art encoder that provides better encoder representations. The input representation for the task of D2T generation is very important to ensure that the generated text perfectly matches with the given input. An example is shown in Table 4.4 where all the models miss some information from the input and/or include wrong information except for BERT2BERT which includes all the given content.

Another important observation is related to GPT-2's suboptimal performance. The intuition that using GPT-2 model as the decoder, along with BERT encoder, for better text generation did not stand. Although BERT2GPT achieves a decent fluency score of 3.691, the model performs poorly on content selection with a faitfluness score of 2.782. It seems that GPT-2's text generation hinders the content selection provided by BERT and the results show that initializing the decoder part with a pre-trained GPT2 checkpoint is not more effective than initializing it with a pre-trained BERT checkpoint even though GPT2 is more similar to the decoder in its architecture. One reason for this would be that it is often beneficial for the encoder and decoder parts of the model to share weights, especially if the target distribution is similar to the input distribution. This is valid in our case since the target text has to perfectly match the input data. As shown in Table 4.4, BERT2GPT misses out on three details present in the input and includes two wrong pieces of information.

The results also validate the importance of pre-training on large corpus with limited resource languages. All the pre-trained models of AraBERT, AraGPT2, and mT5 achieve better performance than the baseline models in both quantitative and qualitative evaluation metrics. The pre-trained language models provide good representations of the Arabic language as they are pre-trained on large Arabic corpus. When fine-tuned, these models need only to learn how to transform the input data into text. As opposed to the baseline model, where the model has to learn both the language representations and how to properly select the content from a small dataset. This, in a way, mimics the process of pipeline-based approaches where content selection is separated from linguistic realization. In general, pre-training improves generalizability and enables the models not to overfit on specific instances in the dataset which in turn helps content selection. This can be seen in the example in Table 4.4 where the baseline model includes information that is not present in the input but is actually correct and present in other instances of the data related to "Adam Cook".

Based on the obtained results, mT5 is the second best performing among the evaluated models which shows that text-to-text pre-training is suitable for the task of D2T generation. One main reason that could lead to mT5 not outperforming BERT2BERT is that it is pre-trained on multilingual and not monolingual Arabic corpus. This is noticed in the literature [54] where multilingual pre-trained language models lag behind their monolingual competitors. Due to the constant number of parameters, the model capacity for rich-resource languages decreases if we have a lot of languages in the pre-training process. Another thing to note is that AraBERT is pre-trained in a self-supervised fashion on text extracted from Arabic news. News articles include similar semantics and structure to the AraWebNLG dataset which mainly includes factual news-like sentences. This helps achieving high text fluency for the BERT2BERT model.

Finally, the dataset includes a lot of proper nouns and relationships between them. Because the dataset is translated from English, these nouns are transcribed to Arabic. These proper nouns are very specific to other languages such as English, Spanish, Brazilian, etc. This makes it hard for the models to associate the proper gender pronouns. This is especially true since the pre-trained corpus used to train AraBERT and AraGT2 is taken from Arabic news text where such confusing proper nouns do not exist. As can be seen in Table 4.4, the baseline model mistakenly uses the feminine pronoun when referring to a man. This emphasises the importance of data-centric approaches to ensure that the texts are properly cleaned and that the models are not being fed confusing data.

To compare our models against the state-of-the-art English models, we fine-tune the pre-trained English T5 and the multi-lingual mT5 models on the same subset of the WebNLG dataset, i.e., we only select the 7k instances that were translated out of the total dataset. This eliminates the effect of the size of the dataset and provides a more just comparison. The mT5 model achieves a BLEU score of 17.9 which is lower than the score achieved with the same model for the Arabic language. However, the T5 model pre-trained only on English corpora is able to achieve a 39.57 BLEU score, 13 points higher than the best pretrained Arabic BERT2BERT model but still within the same range of understandable to good outputs. This reinforces the idea that text-to-text pertaining on monolingual Arabic corpora would lead to state-of-the-art results on the Arabic D2T task.

As for the Russian task, pretrained models such as T5 achieve the best performance on this task. The automatic metric evaluation scores achieved are competitive to the English task with a BLEU score of 45 [55] on 15k instances of training data compared to a score of 65 for the T5 model fine-tuned on 25k instances of English data. These numerical results show that more data is always needed to achieve better performance and that a larger Arabic dataset is needed. The authors also go over some failure cases such as the omitting of information for long inputs of 7 RDF triples and failures related to proper nouns that are infrequent in the dataset. The authors also conclude that the distributions of the data in the training would require further investigation to analyze the stated issues to provide additional insight into the outputs of the model. These conclusions align with the observations previously stated in our work.

Input
آدم كوك | معارك | الحرب البولندية السوفيتية
Adam Koc | battles | Polish–Soviet War
حرب البولندية السوفيتية | جزء من صراع | الحرب الأهلية الروسية
Polish–Soviet War | isPart OfMilitary Conflict | Russian Civil War
حرب البولندية السوفيتية | القائد | ليون تروتسكي
Polish–Soviet War | commander | Leon Trotsky
Baseline Model
حارب آدم كوك في الحرب البولندية السوفيتية ، وحصلتعلى جائزة فلايتي ميليتشي
Missing Info: ليون تروتسكي - الحرب الأهلية الروسية
EN: Adam Cook fought in the Polish-Soviet War, and (she) was awarded the Fleetti Milici Award.
BERT2BERT
شارك آدم كوك في معارك في الحرب البولندية السوفيتية التي كانت جزءا من صراع الحرب الأهلية الروسية ، حيث كان ليون تروتسكي قائد
EN: Adam Cook participated in the battles of the Polish-Soviet War that were part of the conflict of the Russian Civil War, with Leon Trotsky as commander.
BERT2GPT
كان جوزيف ستالين قائد في الحرب الأهلية الروسية، حيث كان جيفرسون ديفيس قائد
Missing Info: آدم كوك - ليون تروتسكي - الحرب الأهلية الروسية
EN: Joseph Stalin was a commander in the Russian Civil War , with Jefferson Davis as a commander
mT5
قاتل آدم كوك في الحرب البولندية السوفيتية، حيث كان ليون تروتسكي قائد
Missing Info: الحرب الأهلية الروسية
EN: Adam Cook fought in the Polish-Soviet War, with Leon Trotsky as commander.

Table 4.4: Example of the output text for each model where Red indicates a wrong information and Orange indicates a writing mistake. Missing information is indicated for each model.

Chapter 5

Conclusion

In this paper, we present the first work to address the D2T generation in the Arabic language. This work addresses the challenges of content selection and fluent text generation in Arabic while focusing on the challenge of low resource availability. A new dataset, AraWebNLG, is introduced for the task. This dataset is translated and reviewed from a smaller subset of the English WebNLG dataset. In this work, we focused on end-to-end approaches as they are more efficient than pipeline-based approaches in the English language. We also leverage pre-trained language models including AraBERT, AraGPT2 and mT5 to address the challenge of low resource availability. Our results highlight importance of pre-training on large corpus of Arabic data for the task in terms of fluency and faithfulness. The BERT2BERT model initialized with AraBERT checkpoint outperformed all other models which reflects the importance of having good input encoding representations for the task at hand. Another finding is that it is important for the encoder and decoder to share weights at initialization as the results showed that the AraGPT2 initialized decoder is not suitable for the task. Lastly, we can also conclude that text-to-text pretraining (mT5) is suitable for the D2T generation on Arabic data even with multilingual pre-training. Further work may include the adoption of a data-centric approach to improve common failure cases related to Arabic-specific challenges and presenting an enhanced version of the data. Another area of improvement is the monolingual pre-training for the T5 model on Arabic data which is expected to give competitive results compared to the BERT2BERT model initialized with monolingual AraBERT weights. Another area of research is to introduce additional datasets for the task by extracting Arabic Wikipedia text and mapping them to their corresponding WikiData [56] RDF triples. Although this would require extensive dataset preparation and cleaning, this eliminates the need for translation and ensures that the data is tailored toward the Arabic language.

Bibliography

[1] S. Wiseman, S. M. Shieber, and A. M. Rush, "Challenges in data-to-document generation," *arXiv preprint arXiv:1707.08052*, 2017.

[2] Y. Puzikov and I. Gurevych, "E2e nlg challenge: Neural models vs. templates," in *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 463–471, 2018.

[3] T. C. Ferreira, C. van der Lee, E. Van Miltenburg, and E. Krahmer, "Neural data-to-text generation: A comparison between pipeline and end-to-end architectures," *arXiv preprint arXiv:1908.09022*, 2019.

[4] M. Kale and A. Rastogi, "Text-to-text pre-training for data-to-text tasks," *arXiv preprint arXiv:2005.10433*, 2020.

[5] L. F. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych, "Investigating pretrained language models for graph-to-text generation," *arXiv preprint arXiv:2007.08426*, 2020.

[6] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "The webnlg challenge: Generating text from rdf data," in *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 124–133, 2017.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[8] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," *arXiv preprint arXiv:1804.09849*, 2018.

[9] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[10] A. Fan, M. Lewis, and Y. Dauphin, “Hierarchical neural story generation,” *arXiv preprint arXiv:1805.04833*, 2018.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[14] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[15] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[16] Y.-C. Chen, Z. Gan, Y. Cheng, J. Liu, and J. Liu, “Distilling knowledge learned in bert for text generation,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7893–7905, 2020.

[17] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.

[18] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” in *Advances in neural information processing systems*, pp. 5753–5763, 2019.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[20] P. Budzianowski and I. Vulić, “Hello, it’s gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems,” *arXiv preprint arXiv:1907.05774*, 2019.

[21] J. Guan, F. Huang, Z. Zhao, X. Zhu, and M. Huang, “A knowledge-enhanced pretraining model for commonsense story generation,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 93–108, 2020.

[22] Y. Liao, Y. Wang, Q. Liu, and X. Jiang, "Gpt-based generation for classical chinese poetry," *arXiv preprint arXiv:1907.00151*, 2019.

[23] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[25] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," *arXiv preprint arXiv:2010.11934*, 2020.

[26] S. Rothe, S. Narayan, and A. Severyn, "Leveraging pre-trained checkpoints for sequence generation tasks," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 264–280, 2020.

[27] Y. Zhou, Y. Tsvetkov, A. W. Black, and Z. Yu, "Augmenting non-collaborative dialog systems with explicit semantic and strategic dialog history," *arXiv preprint arXiv:1909.13425*, 2019.

[28] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan, "Style transfer in text: Exploration and evaluation," *arXiv preprint arXiv:1711.06861*, 2017.

[29] W. Wang, Z. Gan, H. Xu, R. Zhang, G. Wang, D. Shen, C. Chen, and L. Carin, "Topic-guided variational autoencoders for text generation," *arXiv preprint arXiv:1903.07137*, 2019.

[30] W. Fedus, I. Goodfellow, and A. M. Dai, "Maskgan: Better text generation via filling in the_," *arXiv preprint arXiv:1801.07736*, 2018.

[31] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[32] A. V. Mota, T. L. C. da Silva, and J. A. F. De Macêdo, "Template-based multi-solution approach for data-to-text generation," in *European Conference on Advances in Databases and Information Systems*, pp. 157–170, Springer, 2020.

[33] A. Moryossef, Y. Goldberg, and I. Dagan, "Step-by-step: Separating planning from realization in neural data-to-text generation," *arXiv preprint arXiv:1904.03396*, 2019.

[34] L. M. Werlen, M. Marone, and H. Hassan, “Selecting, planning, and rewriting: A modular approach for data-to-document generation and translation,” in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 289–296, 2019.

[35] R. Ye, W. Shi, H. Zhou, Z. Wei, and L. Li, “Variational template machine for data-to-text generation,” *arXiv preprint arXiv:2002.01127*, 2020.

[36] R. Puduppully, L. Dong, and M. Lapata, “Data-to-text generation with content selection and planning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 6908–6915, 2019.

[37] C. Rebuffel, L. Soulier, G. Scoutheeten, and P. Gallinari, “A hierarchical model for data-to-text generation,” *Advances in Information Retrieval*, vol. 12035, p. 65, 2020.

[38] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić, “Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling,” *arXiv preprint arXiv:1810.00278*, 2018.

[39] A. P. Parikh, X. Wang, S. Gehrmann, M. Faruqui, B. Dhingra, D. Yang, and D. Das, “Totto: A controlled table-to-text generation dataset,” *arXiv preprint arXiv:2004.14373*, 2020.

[40] J. Kanerva, S. Rönnqvist, R. Kekki, T. Salakoski, and F. Ginter, “Template-free data-to-text generation of finnish sports news,” *arXiv preprint arXiv:1910.01863*, 2019.

[41] M. Cremaschi, F. Bianchi, A. Maurino, and A. P. Pierotti, “Supporting journalism by combining neural language generation and knowledge graphs.,” in *CLiC-it*, 2019.

[42] M. S. H. Ameur, F. Meziane, and A. Guessoum, “Arabic machine translation: A survey of the latest trends and challenges,” *Computer Science Review*, vol. 38, p. 100305, 2020.

[43] S. AlHumoud, A. Al Wazrah, and W. Aldamegh, “Arabic chatbots: A survey,” *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 9, no. 8, pp. 535–541, 2018.

[44] O. ElJundi, M. Dhaybi, K. Mokadam, H. M. Hajj, and D. C. Asmar, “Resources and end-to-end neural network models for arabic image captioning.,” in *VISIGRAPP (5: VISAPP)*, pp. 233–241, 2020.

[45] S. Talafha and B. Rekabdar, “Arabic poem generation with hierarchical recurrent attentional network,” in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pp. 316–323, IEEE, 2019.

[46] A. M. Azmi and N. I. Altmami, “An abstractive arabic text summarizer with user controlled granularity,” *Information Processing & Management*, vol. 54, no. 6, pp. 903–921, 2018.

[47] F. Alotaiby, S. Foda, and I. Alkharashi, “New approaches to automatic headline generation for arabic documents,” *Journal of Engineering and Computer Innovations*, vol. 3, no. 1, pp. 11–25, 2012.

[48] O. ElJundi, W. Antoun, N. El Droubi, H. Hajj, W. El-Hajj, and K. Shaban, “hulmona: The universal language model in arabic,” in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 68–77, 2019.

[49] W. Antoun, F. Baly, and H. Hajj, “Arabert: Transformer-based model for arabic language understanding,” *arXiv preprint arXiv:2003.00104*, 2020.

[50] W. Antoun, F. Baly, and H. Hajj, “Aragpt2: pre-trained transformer for arabic language generation,” *arXiv preprint arXiv:2012.15520*, 2020.

[51] T. Ferreira, C. Gardent, N. Ilinykh, C. van der Lee, S. Mille, D. Moussallem, and A. Shimorina, “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+)*, 2020.

[52] K. Darwish and H. Mubarak, “Farasa: A new fast and accurate arabic word segmenter,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1070–1074, 2016.

[53] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, “Opennmt: Open-source toolkit for neural machine translation,” *arXiv preprint arXiv:1701.02810*, 2017.

[54] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” *arXiv preprint arXiv:1911.02116*, 2019.

[55] X. Li, A. Maskharashvili, S. J. Stevens-Guille, and M. White, “Leveraging large pretrained models for webnlg 2020,” in *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pp. 117–124, 2020.

[56] D. Vrandečić and M. Krötzsch, “Wikidata: a free collaborative knowledgebase,” *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.