EXPLORING THE REFORMULATION OF NLP TASKS AS TEXT GENERATION TASKS

by

RASTI YASEEN HASAN

A THESIS

Presented to the Department of Computer and Information Science
and the Division of Graduate Studies of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 2022

THESIS APPROVAL PAGE

Student: Rasti Yaseen Hasan

Title: Exploring the Reformulation of NLP Tasks as Text Generation Tasks

This thesis has been accepted and approved in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science by:

Thien Nguyen Chairperson

and

Krista Chronister Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded June 2022

THESIS ABSTRACT

Rasti Yaseen Hasan

Master of Science

Department of Computer and Information Science

June 2022

Title: Exploring the Reformulation of NLP Tasks as Text Generation Tasks

In recent years, NLP classification tasks have been reformulated as text generation tasks in the form of text-to-text transformer-based models that achieve state-of-the-art performance by better utilizing pre-trained language models. This work provides a historical background, a taxonomy based on the output structures of these methods, an exploration of aspects of such models with several representative works, and discusses the current state and future of these models.

CURRICULUM VITAE

NAME OF AUTHOR: Rasti Yaseen Hasan

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene
University of Kurdistan Hewlêr

DEGREES AWARDED:

Master of Master of Science, Computer Science, 2022, University of Oregon
Bachelor of Science, Computer Engineering, 2017, University of Kurdistan Hewlêr

AREAS OF SPECIAL INTEREST:

Artificial Intelligence
Machine Learning
Natural Language Processing

PROFESSIONAL EXPERIENCE:

IT Technician, Seeking to Equip People (STEP), 2018-2020

System Developer and Administrator, Hawler Private Hospital, 2018

Software System Engineer, Patterns Lab, 2017-2018

Language Technology Development Assistant, University of Kurdistan Hewlêr, 2016

GRANTS, AWARDS, AND HONORS:

Fulbright Grantee, The Fulbright Foreign Student Program, 2020

PUBLICATIONS:

R. Yaseen and H. Hassani, "Kurdish Optical Character Recognition" *UKH Journal of Science and Engineering*, vol. 2, no. 1, pp. 18-27, 2018.

ACKNOWLEDGMENTS

Thanks to my family for their continued support throughout my studies. Their sacrifices and hard work are the only reasons I have been able to get where I am today.

I express my utmost gratitude to professor Thien Nguyen for guiding me throughout this process. Finally, I would like to give a special thanks to the Fulbright Foreign Student Program. This work and my academic journey would not have been possible without their amazing support.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

Recent works on various Natural Language Processing (NLP) tasks have utilized **P**re-**T**rained language **M**odels (PTMs[1]) such as BERT [1], GPT-2 [2], BART [3], and T5 [4]. PTMs have been shown to be very useful in improving performance on many NLP tasks [5, 6, 7] since models designed for downstream tasks can leverage the latent knowledge PTMs have about language. Language modelling (predicting the likelihood of tokens in text sequences) does not require labeled data, so the main reason for utilizing PTMs is that finding large-scale labeled corpora for any specific task can be rather challenging, while constructing unlabeled corpora to train PTMs is much more of manageable goal [8]. Consequently, models that utilize PTMs are easier to compute, and do not have to be trained on massive amounts of labeled data to achieve good performance. A major caveat that should be noted is that PTMs themselves are much more complex and time-consuming to compute [9].

Many state-of-the-art (SOTA) systems have generally treated NLP tasks such as semantic role labeling and other information extraction subtasks (e.g., named entity recognition, coreference resolution, and event extraction) as classification problems where a discriminative model is trained to identify which labels/classes input sequences belong to. However, it can be difficult for these discriminative models to take full advantage of the benefits that pre-trained models provide through their latent knowledge. Most notably, discriminative models interact with labels as numbers rather than natural language words that can have relevant semantic information about certain tasks [10].

Over the past few years, several works have proposed reformulating these tasks as text-to-text translation systems where the input is transformed into deterministically decodable textual output formats and the task is encoded into a predefined natural language structure. In this new paradigm, a generative transformer-based model is then trained to learn the new representation and extract desired output structures from the augmented text [10, 11, 12]. For brevity, we refer to this paradigm as the "reformulation paradigm" in this work. An advantage of

[1] As pointed out in [8], we refer to "pre-trained language models" as "PTMs" instead of "PLMs" to avoid confusion with "probabilistic language models".

this approach is that the pre-trained models only need to incorporate the new structure to understand what it represents. This process is analogous to a translation or sometimes a summarization objective [13, 14]. Additionally, the labels of sequences are incorporated into the generation task itself whereby PTMs can have a better understanding of what the model is trying to achieve [4, 11, 15]. This incorporation of the label semantics into the prediction task has led to systems becoming very successful at achieving SOTA results on various NLP tasks.

Commonly used PTMs in this paradigm are BART [3] and T5 [4]. These PTMs are sequence-to-sequence encoder-decoder models that can, like BERT, take advantage of the full context in their encoder, but also, like GPT, be used for generation tasks with an autoregressive decoder as shown in Figure 1.

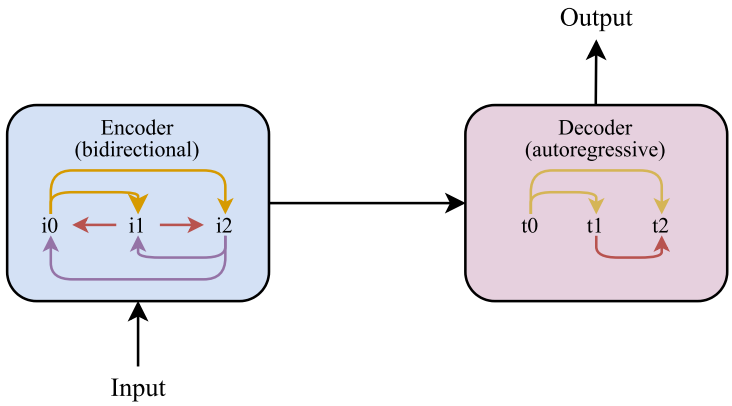

Figure 1. Basic representation of BART and T5. Encoder-decoder with bidirectional encoder and autoregressive decoder.

Systems that reformulate NLP tasks as generation tasks can be divided into four categories based on their output structure:

1- **Augmented text**: Output is a copy of the input text augmented with labels and structure indicators.
2- **Linearized text**: Output is a predefined structure that can be deterministically decoded[2] into task structures.
3- **Templates with placeholders**: Output is a predefined template that implicitly informs the PTM of the types of desired output and contains placeholders for the system to fill with the outputs.

[2] This structure decoding is not to be confused with the decoding process in the transformer decoder.

4- **Index Generation**: Output is word indexes corresponding to, and limited to, words in the input.

TANL [10] (**T**ranslation between **A**ugmented **N**atural **L**anguages) generates *augmented text* named "augmented natural language" for structured prediction tasks where, for each specific task, brackets and vertical bars are used as structure indicators for the model. For example, in semantic role labeling, given an input with a predicate, the output is expected to contain a list of arguments and the semantic role for each argument within said structure indicators. GENRE [14] uses a similar approach to perform entity linking where, given an input text, an augmented output is generated with identified entities and their links as Wikipedia articles.

For event extraction, DEGREE [16] (Data-Efficient Generative Event Extraction) and a model proposed by Li et al. [17] (referred to as "BART-Gen" in this work) generate *templates with placeholders* from which a language model learns the event extraction process. DEGREE and BART-Gen generate whole sentences from templates that correspond to event types (e.g. Justice:Sue, Movement:Transport) [16]. These templates contain placeholders that the model is trained to fill in. The final prediction is then made by comparing the template and output text to extract the spans that were filled in, and search for matches in the original text.

With TempGen [18], instead of creating whole sentences, a *structure linearized text* format is generated for document-level entity-based extraction where several special tags are used as structure indicators to form a template that can be decoded to fill the final extraction template. In GenIE [19], four special tokens are used to indicate the structure of the subject, relation, and object triplets for closed information extraction. REBEL [13] is another model in this family that addresses the joint entity and relation extraction task. This model is similar to the others in terms of its architecture but differentiates itself from both by utilizing a more compact output structure that can represent multiple relations with a single clause.

Other works break the process down into two parts. A PTM is utilized to generate token embeddings with the encoder and the decoder uses a pointer mechanism to *generate the next index* [20, 21, 22]. Yang et. al [20] use this method to tackle flat, nested, and discontinuous named entity recognition. In this work, an input sentence is provided to the model, and the decoder autoregressively generates the index of the next token in the original sentence to generate an output text composed of consecutive entity-tag pairs. For simplicity, we refer to this

work as “BARTNER”. While index generation is not strictly a text-to-text system, these models are architecturally very similar. In fact, the main reason this approach is used instead of direct text generation is to constrain the outputs to valid task structures.

While these new models achieve SOTA or very competitive results on various tasks, it is not very clear what their outstanding performances can be attributed to. Can these results be attributed to the underlying PTMs they use? Are the results reproduceable with different types and sizes of PTMs? How much do their specific output formats contribute to or limit performance?

In the next sections, the NLP task reformulation paradigm is defined and the historical context and progression towards them is highlighted. Next, a discussion of the categorization based on output structures is provided. Then, a discussion of the advantages and disadvantages of these systems is provided. Before concluding, the questions posed here will be analyzed. To summarize, our main contributions in this work are as follows:

1- We provide a detailed definition for the reformulation paradigm and provide a historical context.
2- We provide a taxonomy for this new paradigm based on their output structures.
3- We provide detailed analysis and conduct various tests to identify the reasons for the success of this paradigm.

CHAPTER II

BACKGROUND

Models that use the new paradigm of reformulating or redefining NLP tasks, that have traditionally been framed as classification problems, as generation tasks are generative models that learn deterministically decodable output formats corresponding to target tasks in the training phase and apply that format to new data. In this work, we focus on models that utilize transformer-based PTMs such as BERT [1], BART [3], and T5 [4] since these models outperform traditional convolutional and recurrent neural network based models due to their ability to capture global dependencies [23].

PTMs have been successful primarily because they are transfer learning tools which extract knowledge from one or more tasks/domains and apply it to a target task/domain [8, 24, 25]. Although there are several types of transfer learning such as domain adaptation, cross-lingual learning, and multi-task learning [8], we highlight transfer learning with PTMs and fine-tuning models for target tasks. This concept is the underlying idea that makes this paradigm effective for NLP tasks [10, 26].

A. Language Representation

The main benefit of pre-training is learning universal language representations that will be useful in downstream tasks [27] since better model initialization leads to faster convergence on target tasks [28]. Erhan et al. [29] argue that this is because pre-training is a form of regularization. The initial attempts at representing natural language involved word embeddings. Later, contextual representations were proposed, and they have become the predominant way in which languages are represented [30].

1. Word Embeddings

Representing words as feature vectors has been a common approach where, ideally, a similarity function can accurately capture the syntactic and/or semantic similarity between words through their vector representations. Bengio et al. [31] proposed a method for generating feature vectors using a deep neural network. They called these vectors “distributed representations for

words". Collobert et al. [27] showed that converting words into feature vectors can increase performance on various downstream NLP tasks such as part-of-speech tagging, semantic role labeling, and chunking. These initial findings spawned a great number of subsequent research into different ways of representing language such that they capture useful information about downstream NLP tasks.

Mikolov et al. [32] proposed Continuous Bag-of-Words (CBOW) and Skip-gram for efficient estimation of word representations that did not require deep learning (there are no non-linear hidden layers). CBOW predicts the current word from a window of surrounding words before and after it without regard for the order of words. Conversely, the Skip-gram model predicts a window of surrounding words based on the current word. A simple method for representing phrases, where word order matters, was proposed later for both models as well as other methods to increase their speed and accuracy [33]. These models were the learning mechanisms for Word2Vec. Another popular model in this category is GloVe which is inspired by Skip-gram and learns word representations based on global word-word co-occurrence statistics [34].

While word embeddings can capture semantic and syntactic similarities, they suffer from several limitations:

- They do not capture higher-level linguistic concepts such as polysemous disambiguation, syntactic structures, semantic roles, and anaphora [8].
- Traditional word embedding models like Word2Vec and GloVe, do not have a way to deal with out-of-vocabulary (OOV) words. However, a possible way to handle OOV words is sub-word-level embeddings like fastText [35].
- Reducing words into single points in vector space provides limited semantic understanding [36] which is why several works propose different representations such as Gaussian embeddings [37], hyperbolic space embeddings [38], and multimodal embeddings [39].

2. Contextual Representations

To solve some of the issues inherent to word embeddings, several works started incorporating representations from the hidden layers of deep learning models to contextualize text representations.

Dai and Le [5] proposed two recurrent models. The first was a Recurrent Neural Network (RNN) based language model [40] that was trained to predict the next word based on previous words in a sequence. The second model was a sequence autoencoder inspired by the sequence-to-sequence (Seq2Seq) encoder-decoder LSTM from Sutskever et al. [41], utilized as an unsupervised model whose purpose was to reconstruct the input sequence itself. A supervised LSTM model was tested with random initialization and the two recurrent models: Language Model initialization (LM-LSTM) and Sequence Autoencoder initialization (SA-LSTM). LM-LSTM and SA-LSTM both outperformed random initialization and were either competitive with or outperformed previous work in sentiment analysis (SA) on IMDB [42] and Rotten Tomatoes [43] data sets, as well as text classification tasks on 20 Newsgroups [44] and DBpedia [45] (character-level) data sets.

McCann et al. [28] showed that using the output of Seq2Seq LSTM encoders pre-trained on machine translation data sets, can improve performance on downstream tasks such as sentiment analysis, question answering, entailment, and classification (specifically, question classification). They called these output embeddings **Co**ntext **Ve**ctors (CoVe). Peters et al. [6] further improved the performance of contextual representations with ELMo (**E**mbeddings from **L**anguage **Mo**dels) by combining the internal states (i.e., combining word representations at each layer) of a bidirectional LSTM. Their approach was inspired by research that suggests different layers of bidirectional RNNs and LSTMS encode different types of information [46, 47].

The generations of GPT [2, 7, 48] and BERT [1] have shown that pre-trained transformer-based models are very effective at capturing context since they are not limited to a shorter range compared to LSTM-based models [7]. The main architectural difference between GPT and BERT is that GPT models are all autoregressive (i.e., only attend to previous context by masking out future context), while BERT performs masked language modelling where attention is bidirectional. Other PTMs can have different training objectives and architectures that are not addressed in this work.

Even though transfer learning with fine-tuned PTMs have shown great results, there are two issues with this approach:

- Different NLP tasks have different output classes, which limits generalizability over multiple tasks [8, 10, 26, 49]. For example, in sentiment analysis, a typical approach is to have a binary classifier signifying a positive or negative sentiment in the input. On the

other hand, in the entailment task, a multi-class model will classify input as “entail”, “contradict” or “neither” [26].

- PTMs do not have semantic knowledge of the labels (i.e., classes are represented as numbers that correspond to a dictionary of labels). For example, in the sentence “George R. R. Martin’s first novel, Dying of the Light, was published in 1977.”, for the joint entity and relation extraction task (Figure 2), if the model had knowledge that the “person” entity can write a “book”, learning the “author” relation could have been easier [10].

Raffel et al. [4] proposed a text-to-text framework that is closely related in concept to the new paradigm of reformulating or redefining NLP tasks as generation tasks. Their unsupervised pre-trained T5 (**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer) model can be considered as part of the family of new models that utilize this new paradigm when it is fine-tuned on downstream tasks. This approach alleviates the issues with traditional PTMs, but since T5 only differentiates between tasks with prefixed inputs (e.g., summarize: *text*, translate from German to English: *text* for summarization and translation tasks respectively), it is not clear how it can be trained for tasks, such as named entity recognition and event extraction, that require rich output structures to encapsulate necessary information [11, 26].

This paradigm is part of a larger movement that cast NLP tasks that have traditionally been approached with discriminative frameworks into a common format or task to facilitate multi-task learning. Examples include unifying tasks as question answering [50] or span extraction [51]. Generative models have shown great promise in this regard [10, 48].

B. Motivations for Utilizing the Reformulation Paradigm

Due to their reliance on PTMs, models in this new paradigm have three important and unique features:

1. They are data-efficient since they attempt to fully utilize the pre-existing language understanding of LMs.
2. They have very expressive output format structures and can encapsulate a variety of potentially desired output structures for different tasks with relative ease compared to discriminative models.

3. They are able to better capture inter-dependencies between tasks during training. For example, on joint entity and relation extraction, entity labels facilitate better relation extraction (see Figure 2).

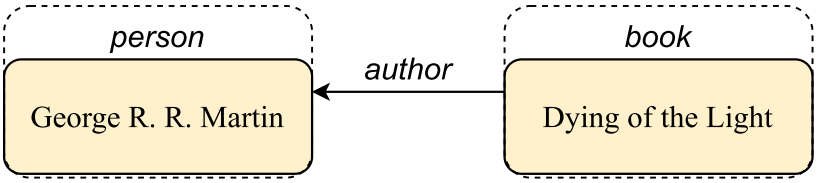

Figure 2. Example of joint entity and relation extraction demonstrating the relevance of the semantic meaning of entity labels to relation extraction.

For the rest of this section, we detail how these features are the reasons that have motivated research into this new paradigm.

1. Low-Resource Settings

The majority of NLP research is concerned with a handful of languages that enjoy an abundance of data, including high-quality annotated data for many NLP tasks. That said, there are several domains and tasks that do not have a great deal of high-quality data even in very data-rich languages such as English. A way to deal with this issue is though transfer learning and, in NLP, a popular transfer learning technique is to use PTMs [52]. Additionally, PTMs are helpful in low-resource languages where unlabeled data is available, but high-quality annotated data for specific NLP tasks are not [25].

Relying on PTMs that have been trained on massive amounts of unlabeled data results in much greater data-efficiency on downstream tasks. Text2Event [11], TANL [10], and DEGREE [16] are all able to perform well in low-resource settings. For example, TANL achieves SOTA in few-shot relation classification on FewRel data set. It also outperforms previous SOTA on CoNLL04 in low-resource setting where only 0.8% to 6% of the training data is used. DEGREE shows that a great deal of the success of the model is due to its ability to make use of label semantics which is also the reason that the model performs well in zero-shot and few-shot scenarios.

2. Multi-Task Learning

Multi-task learning is a very popular field in NLP and machine learning as a whole. This field studies ways to improve both model performance and generalizability. The success of transformer based PTMs has generated more interest in this field among NLP practitioners [4]. One of the reasons for the construction of the T5 model was to enhance the ability of users to perform multiple tasks with the same framework without the need for careful input and output engineering. Furthermore, due to the incredible parameter scaling of T5, it achieved SOTA on multiple tasks using the "pre-train then fine-tune" paradigm without any specific architectural design.

Work in this new paradigm borrows the text-to-text design of T5 and adds more specific and richer output structures to maintain some of the multi-task learnability and improve on SOTA in more specific areas. TANL is the most comprehensive model in this domain that utilizes T5 and achieves SOTA on multiple structure prediction tasks while enjoying an incredible ability to generalize over many tasks. In fact, TANL has demonstrated that fine-tuning on multiple tasks at once increases the performance on many tasks [10]. This shows that models in this paradigm can capture task inter-dependencies.

CHAPTER III

TAXONOMY OF THE PARADIGM

In this section, we provide a taxonomy of models that utilize this approach based on their output structures inspired by Min et al. [52]. For certain models and tasks, the input can also have specific structures. **Table 1** provides a summary of the models described in this section along with some examples.

Table 1. Models in the reformulation paradigm.

	Model	Task(s)	Example	
			Input	Output
Augmented Text	Athiwaratkun et al. [26]	Named Entity Recognition	George R. R. Martin's first novel, Dying of the Light, was published in 1977.	[George R. R. Martin | person]'s first novel, [Dying of the Light | book], was published in 1977.
		Slot Filling	Find me a movie by Steven Spielberg	((FindMovie)) Find me a [movie | genre] by [Steven Spielberg | directed by]
	TANL [10]	Joint Entity and Relation Extraction	George R. R. Martin's first novel, Dying of the Light, was published in 1977.	[George R. R. Martin | person]'s first novel, [Dying of the Light | book | author = George R. R. Martin], was published in 1977.
		Named Entity Recognition	George R. R. Martin's first novel, Dying of the Light, was published in 1977.	[George R. R. Martin | person]'s first novel, [Dying of the Light | book], was published in 1977.
		Relation Classification	[Ramon] , [21], excelled in the prestigious pilot training course, the military said. The relationship between [Ramon] and [21] is	relationship between [Ramon] and [21] = age
		Semantic Role Labeling	The situation on our side and the enemy's side [was] intertwined.	[The situation on our side and the enemy's side | ARG1] was intertwined.
		Event Extraction	**Trigger detection input:** Two soldiers were attacked and injured yesterday. **Argument extraction input (1):** Two soldiers were [attacked | attack] and injured yesterday. **Argument extraction input (2):** Two soldiers were attacked and [injured | injury] yesterday.	**Trigger detection output:** Two soldiers were [attacked | attack] and [injured | injury] yesterday. **Argument extraction output (1):** [Two soldiers | individual | target = attacked] were attacked and injured [yesterday | time | attack time = attacked]. **Argument extraction output (2):** [Two soldiers | individual | target = injured] were attacked and injured [yesterday | time | attack time = injured].
		Coreference Resolution	Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.	[Barack Obama] nominated [Hillary Rodham Clinton] as [his | Barack Obama] [secretary of state | Hillary Rodham Clinton] on Monday. [He | Barack Obama] chose [her | Hillary Rodham Clinton] because [she | Hillary Rodham Clinton] had foreign affairs experience as a former [First Lady | Hillary Rodham Clinton].

Table 1. (continued).

	Model	Task(s)	Input	Output
		Dialogue State Tracking	[user] : I am looking for a place to stay that has cheap price range it should be in a type of hotel [agent] : okay, do you have a specific area you want to stay in? [user] : no, i just need to make sure it s cheap. oh, and i need parking	[belief] *hotel area* not given, *hotel book day* not given, *hotel book people* not given, *hotel book stay* not given, *hotel internet* not given, *hotel name* not given, *hotel parking* yes, *hotel price range* cheap, *hotel stars* not given, *hotel type* hotel [belief]
	GAS-ANNOTATION [53]	Aspect Opinion Pair Extraction	Salads were fantastic, our server was also very helpful.	[Salads | fantastic] were fantastic here, our [server | helpful] was also very helpful.
		Unified Aspect-based Sentiment Analysis	Salads were fantastic, our server was also very helpful.	[Salads | positive] were fantastic here, our [server | positive] was also very helpful.
		Aspect Sentiment Triplet Extraction	The Unibody construction is solid, sleek and beautiful.	The [Unibody construction | positive | solid, sleek, beautiful] is solid, sleek and beautiful
		Target Aspect Sentiment Detection	A big disappointment, all around. The pizza was cold and the cheese wasn't even fully melted.	A big disappointment, all around. The [pizza | food quality | negative] was cold and the [cheese | food quality | negative] wasn't even fully melted [null | restaurant general | negative].
	GENRE [14]	End-to-End Entity Linking	SOCCER − RESULT IN SPANISH FIRST DIVISION. MADRID 1996−08−31 Result of game played in the Spanish first division on Saturday: Deportivo Coruna 1 Real Madrid 1.	SOCCER − RESULT IN [SPANISH] (Spain) FIRST DIVISION [MADRID] (Madrid) 1996−08−31 Result of game played in the [Spanish] (Spain) first division on Saturday: Deportivo Coruna 1 [Real Madrid] (Real Madrid C.F.) 1.
Linearized Text	Text2Event [11]	Event Extraction	Two soldiers were attacked and injured yesterday.	((attack attacked (individual Two soldiers) (time yesterday)) (injury injured (individual Two soldiers) (time yesterday)))
	REBEL [13]		"This Must Be the Place" is a song by new wave band Talking Heads, released in November 1983 as the second single from its fifth album "Speaking in Tongues"	<triplet> This Must Be the Place <subj> Talking Heads <obj> performer <subj> Speaking in Tongues <obj> part of <triplet> Talking Heads <subj> new wave <obj> genre <triplet> Speaking in Tongues <subj> Talking Heads <obj>performer
	TempGen [18]	Role-filler Entity Extraction	Two U.S. mormon missionaries -- aged 19 and 21 -- were shot to death last night by a group of terrorists from the Zarate Wilka Armed Forces of Liberation (FAL). ... blew up the lines providing power to La Paz, ... the U.S. citizens -- Todd Ray Wilson Burdenson and Jeffrey Brent Ball -- .... they were killed with two bursts of machinegun fire. ...	<SOT><SOSN>PerpInd<EOSN><SOE>group of terroists<EOE><SOSN>PerpOrg<EOSN><SOE>Zarate Wilka Armed Forces of Liberation<EOE>...<SOSN>Weapon<EOSN><SOE>machinegun<EOE><EOT>
		Relation Extraction	*Introduction*: Natural language inference (NLI) is an important and significant task in natural language processing (NLP) ... Method: We ... denote the modified ESIM as aESIM... Experiments: The accuracy (ACC) of each method is measured by the commonly used precision score ... It also achieved 88.01 % on Quora ...	<SOT><SOSN>Task<EOSN><SOE>Natural Language Inference<EOE><SOSN>Method<EOSN><SOE>aESIM<EOE><EOT><SOT><SOSN>Material<EOSN><SOE>Quora<EOE><SOSN>Metric<EOSN><SOE>accuracy<EOE><EOT>
	GAS-EXTRACTION [53]	Aspect Opinion Pair Extraction	Salads were fantastic, our server was also very helpful.	(Salads, fantastic); (server, helpful)
		Unified Aspect-based Sentiment Analysis	Salads were fantastic, our server was also very helpful.	(Salads, positive); (server, positive)
		Aspect Sentiment Triplet Extraction	The Unibody construction is solid, sleek and beautiful.	(Unibody construction, solid, positive); (Unibody construction, sleek, positive); (Unibody construction, beautiful, positive);

Table 1. (continued).

Model		Task(s)	Example	
			Input	Output
		Target Aspect Sentiment Detection	A big disappointment, all around. The pizza was cold and the cheese wasn't even fully melted.	(pizza, food quality, negative); (cheese, food quality, negative); (null, restaurant general, negative);
Template Filling	DEGREE [16]	Event Extraction	**Input 1:** Two soldiers were attacked and injured yesterday. [SEP] The event is related to conflict and some violent physical act. [SEP] Similar triggers such as war, attack, terrorism [SEP] Event trigger is <Trigger> [SEP] some attacker attacked some facility, someone, or some organization by some way in somewhere at some time.[3] **Input 2:** ... [SEP] … life and someone is injured. [SEP] … injure, wounded, hurt [SEP] Event trigger is <Trigger> [SEP] somebody or some organization led to some victim injured by some way in somewhere at some time.	**Output 1:** Event trigger is attacked [SEP] some attacker attacked Two soldiers by some way in somewhere at yesterday.[3] **Output 2:** Event trigger is injured [SEP] somebody or some organization led to Two soldiers injured by some way in somewhere at yesterday.
	BART-Gen [17]	Document-Level Event Argument Extraction	<s> <arg1> bought, sold, or traded <arg3> to <arg2> in exchange for <arg4> for the benefit of <arg5> at <arg6> place <s></s> Elliott testified that on April 15, McVeigh came into the body shop and <tgr> reserved <tgr> the truck, to be picked up at 4pm two days later. Elliott said that McVeigh gave him the $280.32 in exact change after declining to pay an additional amount for insurance. Prosecutors say he drove the truck to Geary Lake in Kansas, that 4,000 pounds of ammonium nitrate laced with nitromethane were loaded into the truck there, and that it was driven to Oklahoma City and detonated. </s>	Elliott bought, sold or traded truck to McVeigh in exchange for $280.32 for the benefit of <arg> at body shop place.
Index Generation	BARTNER [20]	Named Entity Recognition	<s> The Lincoln Memorial </s>	123524 Decoded: The Lincoln Memorial <dis> Lincoln </s>
	BARTABSA [21]	Aspect Term Extraction	The wine list is interesting and has good values, but the service is dreadful.	1, 2, 12, 12, </s>
		Opinion Term Extraction		4, 4, 7, 8, 14, 14, </s>
		Aspect-level Sentiment Classification		1, 2 , POS, </s> 12, 12, POS, </s>
		Aspect-oriented Opinion Extraction		1, 2, 4, 4, 7, 8, </s> 12, 12, 14, 14, </s>
		Aspect Term Extraction and Sentiment Classification		1, 2, POS, 12, 12, NEG, </s>
		Pair Extraction		1, 2, 4, 4, 1, 2, 7, 8, 12, 12, 14, 14, </s>
		Triplet Extraction		1, 2, 4, 4, POS, 1, 2, 7, 8, POS, 12, 12, 14, 14, POS, </s>
	Rongali et al. [22]	Slot Filling	Play top hits country	PlayMusicIntent SortType(@ptr1)SortType MediaType(@ptr2)MediaType

[3] The original templates in DEGREE do not contain a time aspect. We have added this part for illustration.

We define the paradigm as techniques that take, as input, natural language text, and have, as their output, decodable task specific structures. For models in this paradigm, the direct output is made up of structured text that can be deterministically decoded into target task structures apart from models in the *index generation* category where the output is indexes decoded into the target task structure. For example, works by Yan et al. [20] and Rongali et al. [22] generate word indexes for aspect-based sentiment analysis and semantic parsing respectively. These methods are architecturally very similar as they all utilize PTMs for their generative prediction process (Figure 3).

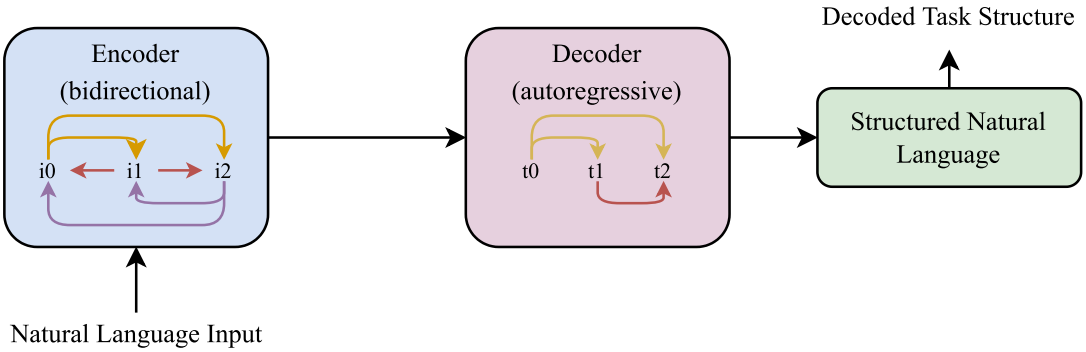

Figure 3 - General output structure of models in the reformulation paradigm. Natural language input is provided to a bidirectional transformer-based encoder, and an autoregressive decoder outputs structured natural language. Natural language input could include triggers for tasks such as event triggers for event argument extraction or appended/prepended with predefined templates. Structured natural language output can be some augmented form of the input or a predifined structure that encapsulates the target task output. For index generation, this block is a set of indexes representing structured natural language. Final decoded task structure is used to detrmine the efficacy of the model.

It should be noted that certain NLP tasks such as machine translation and text summarization are already text generation tasks, but this work focuses only on works that reformulate discriminative tasks as text or index generation. That said, language models have been shown to learn text generation tasks without any supervision [2]. Hence, it could be hypothesized that reformulating discriminative tasks as generation tasks could also benefit from the prior language understanding of language models. To this end, the reformulation process of a discriminative task into a generative task is a three-step process:

1. Given an input sequence x, design an output sequence y such that it includes the necessary information about desired labels for the input sequence.
2. Train a PTM to generate y conditioning on the input x; modeling $P(y|x)$.
3. Decode y to retrieve the desired labels for a given task.

The last step in the above process is in the form of an algorithm whose purpose is to extract the final desired task structures. To retrieve these structures, it is essential that y be designed such that labels are decodable in a deterministic manner.

In the following subsections, models are divided into four categories based on their output structures: augmented text, linearized text, template filling, and index generation.

A. Augmented Text

The output structure of models that use this method is a copy of the input text augmented with labels, structure indicators, and parts of the input text (when tokens are related in some manner). The reason that all input text is copied is that it improves performance and reduces ambiguities when the input contains multiple instances of the same entity [10].

Augmented text outputs were first explored by Athiwaratkun et al. [26] on named entity recognition as well as slot labeling and intent classification tasks. TANL [10] extends this to various structured prediction tasks. Both these works utilize T5 PTMs to generate the augmented text. To decode this augmented text structure, they use the Needleman-Wunsch dynamic programming (DP) based alignment algorithm to identify tokens that match the input text. This strategy has the added benefit of being able to correctly identify slightly misspelled words in the generated augmented text, which their ablation studies show to be very beneficial to the overall performance of their model.

GENRE [14] is a model that performs entity retrieval in an autoregressive manner. It is used for entity disambiguation, document retrieval, and end-to-end entity linking. For the disambiguation and retrieval tasks, the output is only a set of candidate titles from the knowledge base. For entity linking, the output is an augmented representation of the input text where mentions followed by their links, the title in the knowledge base, are each enclosed using special tokens. GENRE ensures that the generated output only contains valid entities using a constrained beam search strategy where a trie specifies all the possible continuations conditioned on the tokens generated prior to the next step.

GAS [53] applies a TANL-like format to aspect-based sentiment analysis. Instead of using DP alignment, this model uses a prediction normalization strategy where a list of valid outputs is constructed corresponding to the input text and labels for each subtask tackled in the

work. Then, if a predicted token does not correspond to any valid output, it is transformed into a valid output token that has the smallest Levenshtein distance to it. In addition to this approach that they call “annotation-style”, they also use an “extraction-style” which is a structure linearized text output with aspect, opinion, and sentiment polarity triplets similar to approaches discussed in section **B** below.

B. Linearized Text

Tasks that have more complex prediction structures such as event extraction and joint entity and relation extraction often require decomposing the task into multiple subtasks that are predicted and combined to create the finalized output structure [54, 55]. However, these approaches suffer from two issues: (1) they need annotated data for each subtask, and (2) they suffer from error propagation from one subtask to another [11]. To solve these issues, some end-to-end models propose using a linearized output structure that corresponds to the whole task structure.

In Text2Event [11] The decoder is trained to predict the output (a linearized event structure) for event extraction with decoder state in cross-attention with encoder state (as is the case with traditional encoder-decoder machine translation). The linearized event structure is different from a natural sentence structure. The output does not follow syntax constraints of regular sentences and contains many "(" and ")" as structure indicators that do not appear as frequently in other contexts. So, curriculum learning is employed to mitigate this issue. The model is first pre-trained on a simpler similar task (only predicts label and span), and it is trained on the actual task afterwards. This approach significantly improves the model’s accuracy. To construct a linearized format from an event record, the record is first converted into specially constructed tree structures, and depth-first traversal is used (to linearize the tree) such that at each level of depth, the order of the linearized event structure is the same order in which token spans appear in the input. Like GENRE, this model uses trie based constrained decoding.

REBEL [13] is architecturally similar to Text2Event. However, this model does not use any strategy to deal with issues observed in Text2Event when training on complex and highly structured tasks. Instead, they pre-train their model on a large dataset called the REBEL dataset. The model significantly outperforms TANL and other SOTA systems only when it is trained on the REBEL data set.

TempGen [18] analogizes the linearized text outputs to summarization. Using this intuition, the authors of this work enhance the ability of TempGen's underlying BART PTM to identify the most salient input tokens to be outputted by the decoder using a cross-attention guided copy mechanism. Copy mechanisms allow summarization models to copy salient words from the input into the output [56]. With their mechanism, only the Top-*k* attention heads with the greatest significance scores are used to compute the final probability of a word in the decoder. This approach allows their model to achieve SOTA on document-level entity-based extraction tasks.

C. Template Filling

Structured prediction tasks can be formulated as templates. In this strategy, the model has templates for each target structure and fills in the necessary information based on the input text.

DEGREE [16] utilizes a predefined end-to-end template that contains special tokens to be predicted by the decoder. For event detection, the template only consists of Event trigger is <trigger> where <trigger> is the special token to be replaced by the correct sequence in the input text. For event argument extraction, custom templates are constructed based on the event type and contain placeholders that are to be replaced by the model with event arguments extracted from the input text. These two tasks can be combined into event extraction where the template is the combination of the event detection template followed by the argument extraction template. To decode final predictions from the output, the replaced placeholders are detected by comparing the raw template to the model output. These text spans are then tokenized along with the input text to detect the matching positions in the original input text.

Li et al. [17] propose a very similar document-level event argument extraction method. BART-Gen defines a template with event argument placeholders for any given event type and prepends it to the raw input text. To reduce argument type mismatch, the template is appended with "clarifications" for each argument (<arg> is a <type>). The PTM is then used to rank the predicted arguments rather than using a greedy approach. This model and DEGREE both utilize BART as their underlying PTM.

D. Index Generation

Ensuring that generative models do not output invalid identifiers is a challenge in the reformulation paradigm. Even though some models have competitive results with free generation, we have discussed several strategies that either ensure valid identifier decoding or try to mitigate and enhance the model's ability to generate the least number of invalid ones possible. An additional strategy that ensures valid generation is to decode token indexes rather than actual text.

BARTNER [20] and BARTABSA [21] implement this approach to reformulate named entity recognition and aspect-based sentiment analysis tasks into generation tasks respectively. Both systems use an encoder to generate token embeddings and a decoder that utilizes a pointer mechanism to generate entity and label indexes autoregressively. Pointer indexes are then converted back to the tokens in their respective index in the original sentence and tag indexes are converted back to the corresponding token(s) in the task specific label list.

Rongali et al. [22] propose a similar approach for task-oriented semantic parsing where a BERT encoder creates input token embeddings. The target output of this model is a series of slots prepended by an intent type. Each slot starts and ends with the slot name with a series of indexes pointing to the tokens in the input sentence that correspond to the slot type.

CHAPTER IV

EXPLORING THE REFORMULATION PARADIGM

Redefining classification tasks as generation tasks has several unique qualities that require exploration and analysis. We conduct several experiments on different systems to explore this paradigm. Since we conduct these experiments in different environments and sometimes with different model hyperparameters, we provide the default results we have achieved for the models when applicable.

A. Changing Input and Output Formats

One of the main aspects of this paradigm is that the output of the system is natural language text. Many models in this paradigm are trained to translate input text into a predefined output text structure that can be decoded into the desired output structure of the task. These models take advantage of the presence of the target labels using the semantic understanding of PTMs for better predictions. Additionally, special tokens are utilized to specify or separate different parts of the output structure. We explore the impact of changing these structures and label semantics to better understand their impact on the performance of these systems. Specifically, we conduct experiments with TANL [10], REBEL [13], and DEGREE [16].

With TANL, we use a slightly different output format by changing special tokens **[**, |, and **]** to **{**, ~, and **}** respectively to see if the model is sensitive to different structure marker tokens. Additionally, we test the impact of changing entity and relation type names on CoNLL04 data set for joint entity and relation extraction on TANL. Entities and relation types are given their short names rather than their verbose counterparts. Table 2 details the label semantic changes that have been made. On REBEL, we conduct an experiment similar to the first with TANL by changing entity markers for CoNLL04 on relation extraction to natural language names rather than the default data set short tags based on those provided in Table 2. It should be noted that while we do use the short entity labels as special tokens, we are not explicitly performing entity extraction here.

Results on TANL (Table 3) show that using different marker tokens and labels have more, generally negative, impact on entity extraction while, there is little impact on relation

extraction. REBEL seems to have been slightly negatively impacted in relation extraction when using natural entity markers.

Table 2. CoNLL04 transformed entity and relation type labels from natural language to abbreviated labels.

Default natural language labels	Short labels
Entities	
location	Loc
organization	Org
person	Peop
other	Other
Relations	
works for	Work_For
kills	Kill
organization based in	OrgBased_In
lives in	Live_In
located in	Located_In

Table 3. Experiments on TANL (T5 Base) and REBEL (BART Large) with changing output format and labels. Percentage of change in micro-F1 score from baseline for the CoNLL04 data set shows significant negative impact on entity extraction, but smaller impact on relation extraction.

	Experiment	P		R		F1		% Change	
		Ent.	Rel.	Ent.	Rel.	Ent.	Rel.	Ent.	Rel.
TANL	Special tokens **[|]** to **{ ~ }**	90.17	77.17	85.82	67.30	87.94	71.90	-2.65	-0.19
	Entity names	89.47	74.70	89.19	69.23	89.33	71.86	-1.11	-0.25
	Relation names	90.11	73.85	89.76	70.14	89.93	71.93	-0.44	-0.15
	Entity and relation names	89.19	75.39	89.43	68.96	89.31	72.30	-1.13	+0.36
REBEL	Relation names	-	71.68	-	70.44	-	71.06	-	-0.31

Next, we conduct experiments on DEGREE where we provide wrong prompt components in the ACE05-E data set without any additional changes to the end-to-end template. This experiment allows us to get a better understanding of the impact of prompt semantics on predictions. We train the models using the wrong prompts in the training and test them with a set of incorrect prompts consistent with the wrong training prompts. We repeat this process two times. First, we only change the event type definitions by randomly assigning them to an event type in the experiment. Then, we change both the definitions and event keywords. In this setting, the event type definitions and keywords belong to the same event type in the correct setting but assigned to a random event type in the experiment. Providing false prompts to DEGREE significantly diminishes performance on all event extraction criteria even when the test set is consistently assigned the same wrong prompts as shown in Table 4. Interestingly, the model with

wrong event type definitions and keywords performs better overall. This is likely due to the model being less confused when it has more consistent prompts.

Table 4. Experiments on DEGREE with wrong prompt components. Results and percentage of change in micro-F1 scores from baseline for end-to-end model with BART Base on event extraction in ACE05-E shows significant negative impact on performance.

	Changing Event Type Definitions			
	P	**R**	**F1**	**% Change**
Trig-I	86.10	61.20	71.55	-4.56%
Trig-C	83.13	56.78	67.47	-5.17%
Arg-I	69.34	41.69	52.07	-4.23%
Arg-C	66.01	39.59	49.50	-4.22%
	Changing Event Type Definitions and Keywords			
	P	**R**	**F1**	**% Change**
Trig-I	85.36	63.47	72.80	-2.89%
Trig-C	81.89	59.25	68.75	-3.37%
Arg-I	64.71	43.63	52.12	-4.14%
Arg-C	61.03	40.98	49.04	-5.11%

These results on TANL, REBEL, and DEGREE indicate that models are sensitive to the format and semantic meaning of the data. This means that such models can be optimized and tuned using different structures, labels, and prompts. Additionally, each part of the data may require not just specific fine-tuning but also an iterative process where different labels and tokens are tested together for parts of the data to achieve optimal performance.

Finally, another aspect of these models is that they have text as inputs and outputs. This is very significant since many of them require very large input text and generate large outputs which requires more resources. However, longer tokens and output structures may not always lead to better results. For example, using longer, more natural, language for labels in REBEL did not lead to a better performance and using short labels for relation types in TANL lead to better performance on relation extraction with some negative impact on entity extraction. Furthermore, this issue of sequence length is exacerbated when dealing with tasks that inherently require much larger input and output sequences when using some of the models.

An example of a model that requires very lengthy inputs and outputs is TANL, and since it requires that the entire sequence be generated in the output natural language, its performance suffers when applied to document-level tasks such as role-filler entity extraction. To explore the limitations with output sequence lengths, we test TANL on this task on MUC-4 [57] dataset. Following GRIT's [58] formulation, the first occurrence of each entity span is found in a

document and assigned as the golden token spans, but we do not take those whose first occurrence are part of the first word (e.g. if “men” is a mention and the word “government” occurs before “men” in the document). We do not precisely follow GRIT’s evaluation metric for this task. We find the list of predictions that are in the gold labels and only take one correct prediction for each entity and remove all coreferent mentions in the gold labels. We use T5 Base with 512 token sequences. The model does not observe approximately 12.66% of entity occurrences due to document lengths in the dataset. We also experiment with training with 512 input tokens and 1024 output tokens and observe a substantial amount of improvement in model recall due to the better entity coverage as shown in Table 5. Thus, checking the data in terms of the input and output lengths is necessary to achieve reasonable results.

Table 5. TANL on MUC-4 for document-level role-filler event extraction. Results show that short token sequences result in much worse recall.

Output Length	P	R	F1
512	64.26	36.59	46.63
1024	64.47	44.03	52.33

B. Changing PTM Sizes

Since these models take advantage of PTMs for label semantic understanding and data efficiency, we hypothesize that their performances are heavily dependent on the underlying PTMs. To test this hypothesis, we experiment with different sizes of PTMs for TANL, GENRE, REBEL, GenIE, TempGen, BARTNER, and DEGREE. The default PTMs utilized in each work are used to establish the baselines. All models are tested with smaller sized variants of the PTMs except for TempGen where we replace BART Base with BART Large. The results of these experiments are shown in **Table 6**. The models are trained and tested in the following manner:

a- TANL is trained on CoNLL04 (training and development set), GENIA, and CoNLL2012 separately.
b- GENRE is trained on BLINK and tested on AIDA, MSNBC, ACE2004, AQUAINT, CWEB, and WIKI.
c- REBEL is trained on CoNLL04 and NYT separately without fine-tuning on the REBEL data set.

d- GenIE is trained on the Wiki-NRE training set and tested on Wiki-NRE test set and Geo-NRE dataset using the small Wiki-NRE evaluation schema.

e- BARTNER is trained on GENIA and CoNLL2003 separately. On GENIA, we train and test with all three entity representations proposed in the work whereas we only use the *Word* representation for CoNLL2003.

f- DEGREE is trained on ACE05-E data set in the end-to-end configuration.

g- TempGen is trained on MUC-4 and SCIREX separately.

Table 6. Experiments on changing PTM sizes for generative models that reformulate NLP classification tasks into text generation tasks. F1 score pairs are provided for the system's default PTM and the new PTM size.

Default PTM	**New PTM**		**Task**	**Data Set**	**Default F1**	**F1**
T5 Base	T5 Small	TANL	Joint Entity and Relation Extraction	CoNLL04	Ent. 90.33	87.74
					Rel. 72.04	66.34
			Named Entity Recognition	GENIA	76.40	75.21
			Semantic Role Labeling	CoNLL2012	84.96	83.17
BART Large	BART Base	GENRE	Entity Disambiguation	AIDA	76.34	72.42
				MSNBC	77.13	73.93
				ACE2004	74.71	75.1
				AQUAINT	78.95	74.14
				CWEB	63.05	59.23
				WIKI	74.83	71.82
		REBEL	Relation Extraction	CoNLL04	71.28	67.26
				NYT	90.84	89.02
		GenIE	Closed Information Extraction	Wiki-NRE	Micro $89.51_{\pm 0.15}$	$91.05_{\pm 0.15}$
					Macro $40.25_{\pm 1.66}$	$37.09_{\pm 1.55}$
				Geo-NRE	Micro $87.03_{\pm 0.98}$	$89.81_{\pm 0.86}$
					Macro $54.01_{\pm 7.06}$	$51.59_{\pm 6.32}$
		BARTNER *Word*	Nested Named Entity Recognition	GENIA	77.99	78.20
		BARTNER *Span*			78.79	78.56
		BARTNER *BPE*			78.60	76.14
		BARTNER *Word*	Named Entity Recognition	CoNLL2003	93.14	91.77

Table 6. (continued).

Default PTM	New PTM		Task	Data Set	Default F1	F1
		DEGREE	Event Extraction	ACE05-E	Trig-I 75.49	74.97
					Trig-C 71.74	71.15
					Arg-I 56.54	54.37
					Arg-C 54.44	51.68
BART Base	BART Large	TempGen	Role-filler Entity Extraction	MUC-4	56.09	52.69
			Binary Relation Extraction	SCIREX	14.78	11.42

Changing PTM size generally has a negative impact on performance. This is somewhat expected with smaller PTMs since larger PTMs usually have better understanding of language semantics. Different models and data sets within models are not equally affected.

On GENRE, performance decreases for all tested data sets except ACE2004. With BARTNER, there is not much difference between the utilized PTM sizes for *Word* and *Span* representations on GENIA while the smaller PTM leads to worse performance for *Word* representation on CoNLL2003. While there is an improvement in micro-F1 with the smaller PTM on GenIE, macro-F1 is higher in both tests, which means that the larger PTM performs better on less frequent relations in the data sets. We also observe that a larger PTM diminishes performance on TempGen. This is likely because we use the same number of cross-attention heads ($K = 10$) for the larger PTM (with 16 heads rather than 12). Further investigation would be required to find the ideal *K* for BART Large.

These results indicate that while we could generally assume models in this paradigm benefit from the better semantic understanding of larger PTMs, there may be cases where the trade-off between different PTM sizes may lead to choosing smaller variants rather than larger ones.

C. Using Fine-tuned PTMs

BART has been fine-tuned for several downstream tasks using large data sets. Since such variants have different embeddings, we hypothesize that different variants may have an impact on models that rely on them in downstream tasks. We test BARTNER and REBEL with BART

Large CNN/DM and XSum; variants that have been fine-tuned on news summarization. BART Large CNN/DM generates long summaries which are close to the source sentences while BART Large XSum is highly abstractive. We specifically choose to test REBEL since it does not apply any constraint on the output sequence generation. Results for these experiments are presented in Table 7.

Table 7. Experiments on fine-tuned variants of BART Large PTM for REBEL and BARTNER using CoNLL04 and GENIA data sets respectively. Results and percentage of change in micro-F1 from default BART Large are provided.

	Data Set	Fine-tuned Model	P	R	F1	% Change
REBEL	CoNLL04	BART Large CNN/DM	50.81	61.82	55.78	-21.75
		BART Large CNN/DM (Output length: 32)	54.59	65.41	59.51	-16.51%
		BART Large XSum	69.17	67.98	68.57	-3.80
BARTNER (*Word*)	GENIA	BART Large CNN/DM	79.54	76.55	78.02	+0.04
		BART Large XSum	79.51	76.48	77.97	-0.03

Performance on BARTNER does not vary with different underlying PTM variants. This is likely because the encoder is much more relevant in the underlying PTM since the decoder is trained in a completely new manner using the pointer mechanism. With the CNN/DM variant, we see that too many relations are predicted on REBEL. We hypothesize that this is due to the fine-tuning phase that the PTM has had where it was trained to generate sentences similar to those in the input sequence. We test with decreasing the output sequence length for REBEL from 128 to 24 to artificially constrain the model output text. As expected, this results in a higher precision. It is therefore important to analyze the PTMs to be used in this paradigm especially when they are fine-tuned for specific downstream tasks.

CHAPTER V

DISCUSSION AND CONCLUSION

Models in the reformulation paradigm have been shown to be incredibly effective on several classic NLP tasks, achieving SOTA results on numerous data sets. One of the greatest advantages of these models is that they can generalize to several, and sometimes disparate, tasks and subtasks. For example, Text2Event and DEGREE unify event extraction subtasks. Much more broadly, TANL unifies structured prediction tasks. Another significant advantage is that these models can make use of PTMs for effective prediction under few-shot and zero-shot scenarios [16, 26].

However, such models have several shortcomings. The following is a discussion of some of these shortcomings as well as possible ways to address them. They can be summarized into the following three points:

1- The quadratic complexity of transformer based PTMs makes them difficult to compute
2- Due to their generative nature, it is difficult to impose semantic and ontological constraints.
3- These is a lack of standardization in utilizing PTM which makes it difficult to reliably discern which part of the model contributes to better performance.

The models discussed in this survey are all Transformer-based models which have time and space complexities of $O(L^2)$, where L is sentence length [23]. This can limit the use of such models due to memory and time constraints as well as financial and environmental considerations [59]. While certain models such as REBEL specifically try to minimize the input and output sequence sizes, this issue is further amplified with template filling models like DEGREE which appends very lengthy sentences and prompts to the raw input to achieve better data efficiency. However, this limitation is due to the inefficiency of transformers rather than the models themselves and could be improved upon with more efficient transformers such as Reformer [60], Linformer [61], and transformers with clustered attention [62]. That said, to the best of our knowledge, no model in this paradigm has attempted to use efficient transformers as their backbones.

Another concern with utilizing generative models is that the target NLP tasks have specific output structures that cannot be overlooked. Generative models could output a wrong structure or even generate labels that do not exist. Models like Text2Event can prevent such issues more easily since they deal with either a single task or structurally similar tasks where the output can be constrained to only generate valid structures. BART-Gen and DEGREE provide extensive context to the language model to minimize such mistakes. However, these solutions are not feasible for TANL since one of the most important aspects of this model is the fact that it unifies multiple tasks into a single framework. Constraining the structure in one way to ensure validity for a task, may lead to disastrous results for other tasks. This is presumably why TANL does not impose any restrictions on the model to generate valid outputs. Step Decomposed and Constrained Text-to-Text Transformer (SDCT5) [63], a model based on TANL, attempts to address this issue by decomposing the decoding process into three stages such that all tasks share the same structure at the initial stage, and valid tokens could be generated in later stages. While this model does ensure that valid structures are generated, it only outperforms TANL marginally. The author hypothesizes that this is due to exposure bias since output constraints are only applied during inference.

Finally, a more general concern is that these models use different PTMs. Since performance on specific downstream tasks can depend on the choice of PTM [4], it is difficult to analyze whether the remarkable performance achieved by these models reflects the effectiveness of their architectures or if it is due to the outstanding language understanding of a specific PTM, and its fit for the target tasks. A possible way to discern the effectiveness of a specific model architecture would be to investigate its performance without pre-training. Unfortunately, many of the models reviewed in this work do not provide this data. Another possibility would be to train all the models on the same PTM of the same size, but this may sometimes change aspects of the models in undesirable ways since different PTMs have different training objectives and token associations.

In this work, we explored a new paradigm in NLP where tasks are reformulated as text generation tasks. We provided a brief history of the more significant milestones that lead to the emergence and progress of this paradigm. We provided a taxonomy based on the output structures of these works and explored some of the representative models in each type. Finally, we explored several the models to better understand the different aspects of said models in terms

of their output representations, PTM sizes, and PTM variants. We found that these models are sensitive to the specific output formats. Utilizing smaller PTMs generally lead to poorer performance on various tasks. We also found that different models vary in the way they are impacted by PTM variants. Finally, we discussed a number of limitation and possible future directions in this section.

REFERENCES CITED

[1] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, 2019.

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, *Language models are unsupervised multitask learners*, OpenAI Blog, 2019.

[3] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020.

[4] 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," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1-67, 2020.

[5] A. M. Dai and Q. V. Le, "Semi-supervised Sequence Learning," *Advances in Neural Information Processing Systems*, vol. 28, pp. 3079-3087, 2015.

[6] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep Contextualized Word Representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, 2018.

[7] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, *Improving language understanding by generative pre-training*, 2018.

[8] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai and X. Huang, "Pre-trained Models for Natural Language Processing: A Survey," *Science China Technological Sciences*, pp. 1-26, 2020.

[9] E. M. Bender, T. Gebru, A. McMillan-Major and S. Shmitchell, "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Virtual Event, 2021.

[10] G. Paolini, B. Athiwaratkun, J. Krone, A. A. Jie Ma, R. Anubhai, C. N. d. Santos, B. Xiang and S. Soatto, "Structured Prediction as Translation between Augmented Natural Languages," in *9th International Conference on Learning Representations, ICLR 2021*, 2021.

[11] Y. Lu, H. Lin, J. Xu, X. Han, J. Tang, A. Li, L. Sun, M. Liao and S. Chen, "Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

[12] C. Donahue, M. Lee and P. Liang, "Enabling Language Models to Fill in the Blanks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[13] P.-L. H. Cabot and R. Navigli, "REBEL: Relation Extraction By End-to-end Language generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, 2021.

[14] N. D. Cao, G. Izacard, S. Riedel and F. Petroni, "Autoregressive Entity Retrieval," in *International Conference on Learning Representations*, 2021.

[15] W. Zhang, Y. Deng, X. Li, Y. Yuan, L. Bing and W. Lam, "Aspect Sentiment Quad Prediction as Paraphrase Generation," *arXiv preprint*, no. arXiv:2110.00796, 2021.

[16] I.-H. Hsu, K.-H. Huang, E. Boschee, S. Miller, P. Natarajan, K.-W. Chang and N. Peng, "DEGREE: A Data-Efficient Generative Event Extraction Model," in *Arxiv*, 2021.

[17] S. Li, H. Ji and J. Han, "Document-Level Event Argument Extraction by Conditional Generation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, 2021.

[18] K.-H. Huang, S. Tang and N. Peng, "Document-level Entity-based Extraction as Template Generation," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, 2021.

[19] M. Josifoski, N. D. Cao, M. Peyrard, F. Petroni and R. West, "GenIE: Generative Information Extraction," *arXiv preprint arXiv:2112.08340*, 2021.

[20] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang and X. Qiu, "A Unified Generative Framework for Various NER Subtasks," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Online, 2021.

[21] H. Yan, J. Dai, T. Ji, X. Qiu and Z. Zhang, "A Unified Generative Framework for Aspect-based Sentiment Analysis," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

[22] S. Rongali, L. Soldaini, E. Monti and W. Hamza, "Don’t Parse, Generate! A Sequence to Sequence Architecture for Task-Oriented Semantic Parsing," in *Proceedings of The Web Conference 2020*, Taipei, 2020.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[24] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010.

[25] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen and D. Klakow, "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, 2021.

[26] B. Athiwaratkun, C. N. d. Santos, J. Krone and B. Xiang, "Augmented Natural Language for Generative Sequence Labeling," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, 2020.

[27] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research,* vol. 12, pp. 2493-2537, 2011.

[28] B. McCann, J. Bradbury, C. Xiong and R. Socher, "Learned in Translation: Contextualized Word Vectors," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, 2017.

[29] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent and S. Bengio, "Why Does Unsupervised Pre-Training Help Deep Learning?," vol. 11, p. 625–660, 2010.

[30] K. Ethayarajh, "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, 2019.

[31] Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin, "A Neural Probabilistic Language Model," *Journal of Machine Learning Research,* vol. 3, pp. 1137-1155, 2003.

[32] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[33] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, Red Hook, 2013.

[34] J. Pennington, R. Socher and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, 2014.

[35] 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.

[36] S. Ruder, *Word embeddings in 2017: Trends and future directions*, 2017.

[37] L. Vilnis and A. McCallum, "Word Representations via Gaussian Embedding," *ICLR*, 2015.

[38] M. Nickel and D. Kiela, "Poincaré Embeddings for Learning Hierarchical Representations," in *Advances in Neural Information Processing Systems*, 2017.

[39] B. Athiwaratkun and A. G. Wilson, "Multimodal word distributions," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, Vancouver, 2017.

[40] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký and S. Khudanpur, "Recurrent neural network based language model," *Interspeech*, vol. 2, no. 3, pp. 1045-1048, 2010.

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

[42] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, Portland, 2011.

[43] B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, 2005.

[44] K. Lang, "NewsWeeder: Learning to Filter Netnews," in *Proceedings of the 12th International Machine Learning Conference (ML95)*, 1995.

[45] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. v. Kleef, S. Auer and C. Bizer, "DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web,* vol. 6, no. 2, pp. 167-195, 2015.

[46] O. Melamud, J. Goldberger and d. Dagan, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, 2016.

[47] Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad and J. Glass, "What do Neural Machine Translation Models Learn about Morphology?," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, 2017.

[48] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, 2020.

[49] A. C. Stickland and I. Murray, "BERT and PALs: Projected Attention Layers for Efficient Adaptation in Multi-Task Learning," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[50] B. McCann, N. S. Keskar, C. Xiong and R. Socher, "The Natural Language Decathlon: Multitask Learning as Question Answering," *CoRR,* vol. abs/1806.08730, 2018.

[51] N. S. Keskar, B. McCann, C. Xiong and R. Socher, "Unifying Question Answering and Text Classification via Span Extraction," *CoRR,* vol. abs/1904.09286, 2019.

[52] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heinz and D. Roth, *Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey,* arXiv preprint, 2021.

[53] W. Zhang, X. Li, Y. Deng, L. Bing and W. Lam, "Towards Generative Aspect-Based Sentiment Analysis," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021.

[54] T. M. Nguyen and T. H. Nguyen, "One for All: Neural Joint Modeling of Entities and Events," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[55] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, "Relation Classification via Convolutional Deep Neural Network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014.

[56] S. Xu, H. Li, P. Yuan, Y. Wu, X. He and B. Zhou, "Self-Attention Guided Copy Mechanism for Abstractive Summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020.

[57] "Fourth Message Uunderstanding Conference (MUC-4)," McLean, Virginia, 1992.

[58] X. Du, A. Rush and C. Cardie, "GRIT: Generative Role-filler Transformers for Document-level Event Entity Extraction," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021.

[59] E. Strubell, A. Ganesh and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, 2019.

[60] N. Kitaev, L. Kaiser and A. Levskaya, "Reformer: The Efficient Transformer," in *International Conference on Learning Representations*, 2020.

[61] S. Wang, B. Z. Li, M. Khabsa, H. Fang and H. Ma, "Linformer: Self-Attention with Linear Complexity," *CoRR*, vol. abs/2006.04768, 2020.

[62] A. Vyas, A. Katharopoulos and F. Fleuret, "Fast Transformers with Clustered Attention," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[63] X. Cheng, "A deeper look into multi-task learning ability of unified text-to-text transformer," 2021.