Transformer-Based Observers in Psychotherapy

Tarun Sunkaraneni
University of Utah

UUCS-20-011

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

7 August 2020

Abstract

Motivational Interviewing is a style of psychotherapy which has shown success as a method for treating addiction and substance abuse problems. In recent years, Natural Language Processing (NLP) techniques have shown promising results in assisting Motivational Interviewing (MI) training and advancing its popularity. One of the ways they are achieving this is by providing therapists feedback on their counseling efficacy by providing feedback to therapists via automatically tagged Motivational Interviewing skill codes (MISC) of utterances in a session.

The recent transformer architecture-based language models offer off-the-shelf, domain adaptation capabilities and have achieved state-of-the-art results in many NLP tasks. However, these models have not been trained on dialogue data, which is structured much differently than the linear structure of passages. We propose methods to encode dialogue data such that it can be domain adapted for psychotherapy. We achieve state-of-the art results in every measured metric of classifying Motivational Interviewing skill codes. While prior work has sought to model patient and therapist as separate agents, we show that a unified model can further improve state-of-the-art results. We conclude by examining how well classical NLP interpretation methods work on transformer models by analyzing probing results on these models. We are able to show that interpretation techniques, that have been useful for other NLP tasks such as question answering, may not be powerful enough to explain the inner workings of transformer models.

TRANSFORMER-BASED OBSERVERS IN PSYCHOTHERAPY

by

Tarun Sunkaraneni

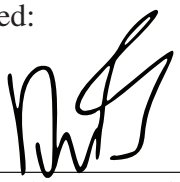
A Senior Honors Thesis Submitted to the Faculty of
The University of Utah
In Partial Fulfillment of the Requirements for the

Honors Degree in Bachelor of Science

In
Computer Science

Jul 31, 2020

Approved:



Vivek Srikumar, PhD Thesis Faculty Supervisor	Mary Hall, PhD Director, School of Computing
Thomas C. Henderson, PhD Honors Faculty Advisor	Sylvia D. Torti, PhD Dean, Honors College

Copyright © Tarun Sunkaraneni 2020
All Rights Reserved

ABSTRACT

Motivational Interviewing is a style of psychotherapy which has shown success as a method for treating addiction and substance abuse problems. In recent years, Natural Language Processing (NLP) techniques have shown promising results in assisting Motivational Interviewing (MI) training and advancing its popularity. One of the ways they are achieving this is by providing therapists feedback on their counseling efficacy by providing feedback to therapists via automatically tagged Motivational Interviewing skill codes (MISC) of utterances in a session.

The recent transformer architecture-based language models offer off-the-shelf, domain adaptation capabilities and have achieved state-of-the-art results in many NLP tasks. However, these models have not been trained on dialogue data, which is structured much differently than the linear structure of passages. We propose methods to encode dialogue data such that it can be domain adapted for psychotherapy. We achieve state-of-the art results in every measured metric of classifying Motivational Interviewing skill codes. While prior work has sought to model patient and therapist as separate agents, we show that a unified model can further improve state-of-the-art results. We conclude by examining how well classical NLP interpretation methods work on transformer models by analyzing probing results on these models. We are able to show that interpretation techniques, that have been useful for other NLP tasks such as question answering, may not be powerful enough to explain the inner workings of transformer models.

CONTENTS

ABSTRACT . . . . . **iii**

LIST OF FIGURES . . . . . **vi**

LIST OF TABLES . . . . . **viii**

NOTATION AND SYMBOLS . . . . . **ix**

CHAPTERS

1. INTRODUCTION **1**

2. BACKGROUND: RECENT ADVANCES IN MODELING LANGUAGE **4**

2.1 Sequence Classification 4
2.2 Word Embeddings 4
2.3 Recurrent Neural Networks 6
2.4 Gated Recurrent Neural Networks 8
2.5 Neural Attention 11
2.6 Transformer Neural Networks 13
2.7 Masked language models: BERT 15

3. MOTIVATIONAL INTERVIEWING **18**

3.1 Motivational Interviewing and Psychotherapy 18
3.2 Natural language processing for MISC 20
3.2.1 MISC Classification Setup 20
3.2.2 Encoding Dialogue 20
3.2.3 Word-level Attention 20
3.2.4 Utterance-level Attention 21
3.2.5 Label Imbalance 21
3.2.6 Results 22

4. APPROACH: REPRESENTING DIALOGUE TO TRANSFORMERS **24**

4.1 Dialogue Encoding 24
4.2 Training and Optimization 25
4.3 Model Interpretation techniques 27

5. RESULTS **28**

5.1 Client Predictions 28
5.2 Therapist Predictions 29
5.3 Unified Predictions 30
5.4 Saliency Maps 30
5.5 Attention Maps 31

6. CONCLUSIONS **34**

REFERENCES **36**

LIST OF FIGURES

2.1 Probabilistic neural model. Figure borrowed from Bengio et at. [6], and highlights the two components f and g that form the language model. The first component converts words $w_{t-n-1}, ...w_{t-1}$ into vectors $C(w_{t-n-1})$, ..., $C(w_{t-1})$ using the lookup table C, the second component is a single layer perceptron function g which outputs probabilities of the next word w_t. 6

2.2 An recurrent neural network. At any timestamp t, input w_t (blue) is converted into e_t and fed into the recurrent cell (green). The recurrent unit uses two activation functions σ_o, σ_h to obtain h_t and o_t(orange). Usually only o_t is used for classification tasks. Usually the sigmoid or softmax activation functions are chosen for σ_o for classification choices, and tanh or sigmoid activation functions are chosen for σ_h to propagate information effectively. . . . 7

2.3 A recurrent neural network language model. At timestamp t, e_t is fed to the recurrent network to attain O_t, which is usually the output of a softmax activation function. Loss $J_t(\theta; O_t, y_t)$ is calculated and backpropagated at every timestamp. 8

2.4 Backpropagation through time. We omit the details of $\frac{\partial J(\theta)}{\partial W_O}$ since it does not backpropagate through time. 8

2.5 Detailed view of the GRU internals. Figure is borrowed from [26]. See text for further details and explanations. 10

2.6 Sequence Modeling tasks. Left: Language modeling is considered a many-to-one sequence learning task, whereas tasks such as machine translation rely on many-to-many classification. Figure is borrowed from [13]. Right: The structure of a sequence-to-sequence classifier. The illustrated model has a unified encoder-decoder RNN, which means that the weights are shared between modules. Usually, a token representing the end of sequence `<eos>` signals the model to start the decoding phase of the task has started. This figure is borrowed from [25]. 11

2.7 A seq2seq attention model. Figure borrowed from Bahadanu et al. [5]. This encoder relies on a Bidirectional-RNN, which consists of concatenating the hidden states of two RNNs, one going forwards, the other going backwards. The attention weights $\alpha_{t,u}$ indicate the amount of attention to pay to encoder output at timestamp u for generating the decoder output at timestamp t. 12

2.8 An attention-based RNN classification model. Whereas earlier models would rely on a combination of pooling over the hidden states and the final hidden state $\mathbf{h}_t$, an attention based model can learn a aggregating function which can be used to classify the sequence. 13

2.9 Left: QKV-self attention described in Vaswani et al. Right: How embeddings x_1, x_2 are converted to their respective queries keys and values. See texts for full details and descriptions. Figures borrowed from [46] and [1]. 14

2.10 BERT model. Figure borrowed from [11]. Pre-trained BERT is trained using 1) Masked language modeling and 2) Next sentence prediction tasks on a large corpus. BERT can then be fine-tuned for other downstream NLP tasks. The `[CLS]` token is a special token added at the beginning of every input example, and is typically used for prediction tasks. `[SEP]` token is used for separating two sentences. This token separates `A` and `B` during pre-training NSP task, but can separate segments such as question-answer pairs during fine-tuning. Figure borrowed from original work. 16

5.1 Saliency map interpretation of a sample utterance on RoBERTa `SPEAKER-SPAN-SEP`. The largest 6 gradients correspond to special tokens, which played an aggregating role during their pre-training and fine-tuning phase. 31

5.2 More gradients for the saliency map interpretation of a sample utterance on RoBERTa `SPEAKER-SPAN-SEP`. The magnitude of token gradients is cryptic and not easily explicable. In this `MIN` example, the word suffer is intuitively a big indicator of the MISC code, but is the eigth most crucial token for the model. 31

5.3 Examples of attention maps from analysis on psychotherapy dialogue data. Only the top 10% attenion connections are shown. Left: typical attention maps in the first layer, shows no real trends. Middle: An attention head responsible for making tokens attend to the next token, with a couple exceptions. It is also from the first layer. Right: An attention head at layer 12, at which point it is difficult to discern the functionality of the attention head. . . . 33

LIST OF TABLES

3.1 Therapist MISC labels and dataset used for the scope of this thesis. Table borrowed from Cao et al. [7] 19

3.2 Client MISC labels and dataset used for the scope of this thesis. Table borrowed from Cao et al. [7] 19

3.3 The word-level attention mechanisms used in Cao et al. [7]. 21

3.4 RNN based MISC classification results on client codes. $\mathcal{C}_C$ is a simple MLP(H_n) + MLP(v_n) model which does not rely on any attention mechanism. 22

3.5 RNN based MISC classification results on therapist codes. Models use the same configuration as Table 3.2, but tuned for therapist anchor utterance codes. 22

4.1 Summary of dialogue data encodings we use in this work. RoBERTa uses `<s>` and `</s>` instead of `[CLS]` and `[SEP]`. It also uses two separator tokens between utterances in place of only one that BERT uses. 25

5.1 Transformer based MISC classification results on client codes. 28

5.2 Transformer based MISC classification results on therapist codes. 29

5.3 Unified Transformer based MISC classification results on patient codes. 30

5.4 Unified Transformer based MISC classification results on therapist codes. 30

NOTATION AND SYMBOLS

$V, \|V\|$	vocabulary and cardinality of a language
w_i	the i-th word in vocabulary, like "apple"
$w_1, ...w_t$	words indexed in a sequence. unrelated to w_i
t, T	sequence position variable, T represents a maximum vaue
$\mathbf{q}$	vector. In general, lower case bold letters
$\mathbf{Q}$	matrix. In general, upper case bold letters
$C(w_i)$	the word embedding of w_i
$\mathbf{e}_t$	the word embedding at timestamp t
$[\mathbf{x}; \mathbf{y}]$	Vector concatenation. if $\mathbf{x} \in \mathbb{R}^a$ and $\mathbf{y} \in \mathbb{R}^b$ then $[\mathbf{x}; \mathbf{y}] \in \mathbb{R}^{a+b}$
$a \circ b$	Hardamard product

CHAPTER 1

INTRODUCTION

Dialogue is the first mode of communication that we learn and develop, yet it is significantly complex due to the volume of external knowledge that is required to comprehend the exchange of information, opinions and intentions in human discourse. Following a conversation requires not only the understanding of individual utterances, but also an understanding of the how these utterances relate to each other, what information can be understood or inferred between the exchanges and how the speaker is an evolving agent throughout the life of the conversation. While great progress has been made in understanding language by attempting to learn model distributed representations and extract contextual embeddings [10], conversations and dialogues often rely on long spans of context, multiple-hops of reasoning, and advanced forms of language inferences [19].

The ultimate goal of dialogue modeling is to produce intelligent agents capable of holding human-like conversations. Conversational agents have been implemented in the field of psychotherapy for a relatively long time, going back to chatbots such as ELIZA [49] or PARRY [9], which aim to simulate an agent conversing with a human participant by relying on pattern matching and rules for generating responses. This thesis explores and studies *observer* agents for psychotherapy which can monitor and moderate a therapy conversation on the fly for a style of therapy known as *motivational interviewing*.

Motivational interviewing (MI) is a style of psychotherapy that is commonly employed to treat patients who suffer from addiction, obesity and other detrimental lifestyle choices [27]. Statements in MI conversations can be categorized using *Motivational Interviewing Skill Codes* (MISC) [18], which bucket each utterance according to its function in the conversation. These codes are used to measure the effectiveness of the technique and the counselor's adherence to the MI principles. They can also be used to evaluate MI training that the counselor has undergone. Some examples of positive MISC labels that are

used to categorize therapist responses are facilitate (FA), giving information (GI), and MI non-adherent behavior (MIN).

MI fidelity and MISC labels are usually annotated by human raters on concluded therapy sessions, a task that has considerable costs associated with time, training, and money [40]. Manual annotation hinders the ability to give a counselor immediate feedback, since it would be useful for a therapist to have access to MISC labels in real-time during a counseling session. Natural language processing and deep learning techniques have recently been utilized for coding MI sessions successfully, but existing works have only relied on recursive or recurrent neural networks [40] [42].

Relevant work has explored the capability of deep learning NLP models which are able to 1) categorize and 2) forecast MISC codes in psychotherapy [7]. Categorization is the task of predicting MISC labels for an agent's most recent utterance given recent dialogue history, while forecasting is the task of predicting the MISC label of the next utterance given the history and the next speaker's identity. We focus entirely on the categorization task for the scope of this thesis, as it is a predecessor of the forecasting task.

The recently proposed transformer neural network architecture [46] is capable of capturing complex sequential information without any locality constraints between the segments of a sequence. These models have revolutionized transfer-learning in natural language processing and usually use a pre-trained transformer model and fine-tune for a specific task [38]. The pre-trained models are general-purpose language models trained on gigabytes of data for days on numerous GPUs [37] and are then domain-adapted for specific use cases. Fine-tuning these models has been very successful, starting with GPT [30] and BERT [11] as they now hold state of the art performances across many prominent natural language processing datasets and tasks such as language modeling [43], question answering [32] and general language understanding [47].

These transformer models are trained using the masked-language modeling (MLM) objective [11] [44], and many of them yield great semantic and syntactic knowledge [31]. However, language modeling does not capture many elements of language that dialogue does, notably the notion of a speaker and the duality of their utterances in a conversation. To this end, we explore the feasibility of domain-adapting these transformer models as dialogue systems. Pre-trained transformer models are usually trained on dumps of the

internet or other media like books, so therapy sessions would very much be a new domain of knowledge for these models.

We start by searching for the best format to encode dialogue data to transformer based language models. Since transformer inputs are not recurrent, they will need to learn sophisticated layouts of information to understand and learn from inflections in conversations that occur due to changing speakers. In order to measure the effectiveness of each input format, we will evaluate the models' ability to classify MISC labels on annotated psychotherapy sessions across different transformer models.

We attempt to understand a model's textual reasoning using gradient-based interpretation methods. We conclude by exploring how a model interprets dialogue by examining a model's self-attention, which can help us understand whether it is possible for these models to learn the notion of dialogue instead of treating it as continuous text. The ability to do so is paramount for processing dialogue data with transformer networks.

This thesis evaluates the effectiveness of training transformer-based observer agents that can automatically label utterances and assist the therapist in providing proper care to their patients. We show that these models can achieve state-of-the-art results with significantly less training steps and can show an understanding of both the patient and therapist in dialogue.

CHAPTER 2

BACKGROUND: RECENT ADVANCES IN MODELING LANGUAGE

In this chapter we provide a brief survey of neural network classifiers for natural language processing that lead up to the transformer neural networks.

2.1 Sequence Classification

MISC prediction, as is dialogue act prediction, is a sequence classification problem. The transformer architecture is derived using a culmination of insights that have revolutionized natural language processing. In this section we will go over advances in NLP which have motivated and paved way for the formulation of the transformer architecture. Each subsection will focus on language modeling, the act of attaining the joint probability of a sentence, $P(w_1, w_2, ..., w_t)$. This formulation is often approached through marginalization,

$$P(w_1, w_2, ..., w_t) = \sum_{w_i \in V} P(w_t = w_i | w_1, w_2, ..., w_{t-1})$$

Because of this decomposition, language modeling can be thought of as deriving the probability of w_t being a certain word $w_i \in V$ given some prior probability $P(w_1, w_2, ..., w_{t-1})$, which is a $|V|$-way classification problem. Sequence classification is an abstraction of this task, in which instead of predicting the next word's identity, we calculate probabilities of higher level concepts such as "Does this sentence contain inflammatory language?" or "Is this paragraph talking about mammals?" Our work focuses on sequence classification, but the advances we highlight are more apparent and motivated through language modeling.

2.2 Word Embeddings

Language is intrinsic and inherent to human understanding, yet is difficult to quantify. This begs the question: what are words, and how do they get their meaning? The first idea to quantify each word is through discrete representations. However this assumption

suffers from the *curse of dimensionality*, where the dimensionality of the resulting representation is computationally infeasible or inefficient. For example, modeling the joint probability of five consecutive words in a language with vocabulary of size 100,000 would result in $100\ 000^5 = 10^{25} - 1$ free parameters. Additionally, an inherent problem with this representation in that these representations have no way of encoding word similarity: it is beneficial to be able to deduce that the representations for "dog" and "cat" line up more than "dog" and "chair."

To capture this phenomenon, distributed probabilistic representations of word vectors were proposed to harness word embeddings in [6]. This work trains a neural model on a sequence $w_1, w_2, ..., w_T$ consisting of words $w_i \in V$, where the vocabulary $\mathbb{R}^V$ is a large but finite set.

The objective was to learn a good model $f(w_{t-n-1}, ..., w_t) = P(w_{t-n-1}|w_1, ..., w_{t-1})$, which is able to generalize to out of sample examples well. The parameter n is the context window size of the model, and is constrained to a small number ($n = 3$ or 5), so f is a probabilistic n-gram model. Model f is composed of two components:

1. A mapping C which converts $w_i \in V$ to a real vector $C(w_i) \in \mathbb{R}^m$. This transformation C is represented by a simple $|V| \times m$ matrix which performs a "lookup" operation. This transformation has since then come to be known as an embedding layer.

2. The probability function, g which maps the words in the n-gram context $[C(w_1); ...; C(w_t)] \in \mathbb{R}^{t \times m}$ and produces $\hat{\boldsymbol{p}}_{\boldsymbol{t}} \in \mathbb{R}^{|V|}$, the conditional probabilities of the next word.

The embedding transformation, C, converts discrete words into real valued vectors and is optimized with n-gram training examples. This allows us to learn distributed representation for words, which leverages word contexts to derive meaning. While it is now possible to represent words in a continuous space, a problem with the function g is that it works over a fixed context window n, which makes it difficult to scale this solution over large contexts such as dialogue. With word vectors established, we now explore a model which is capable of handling sequences of arbitrary length.

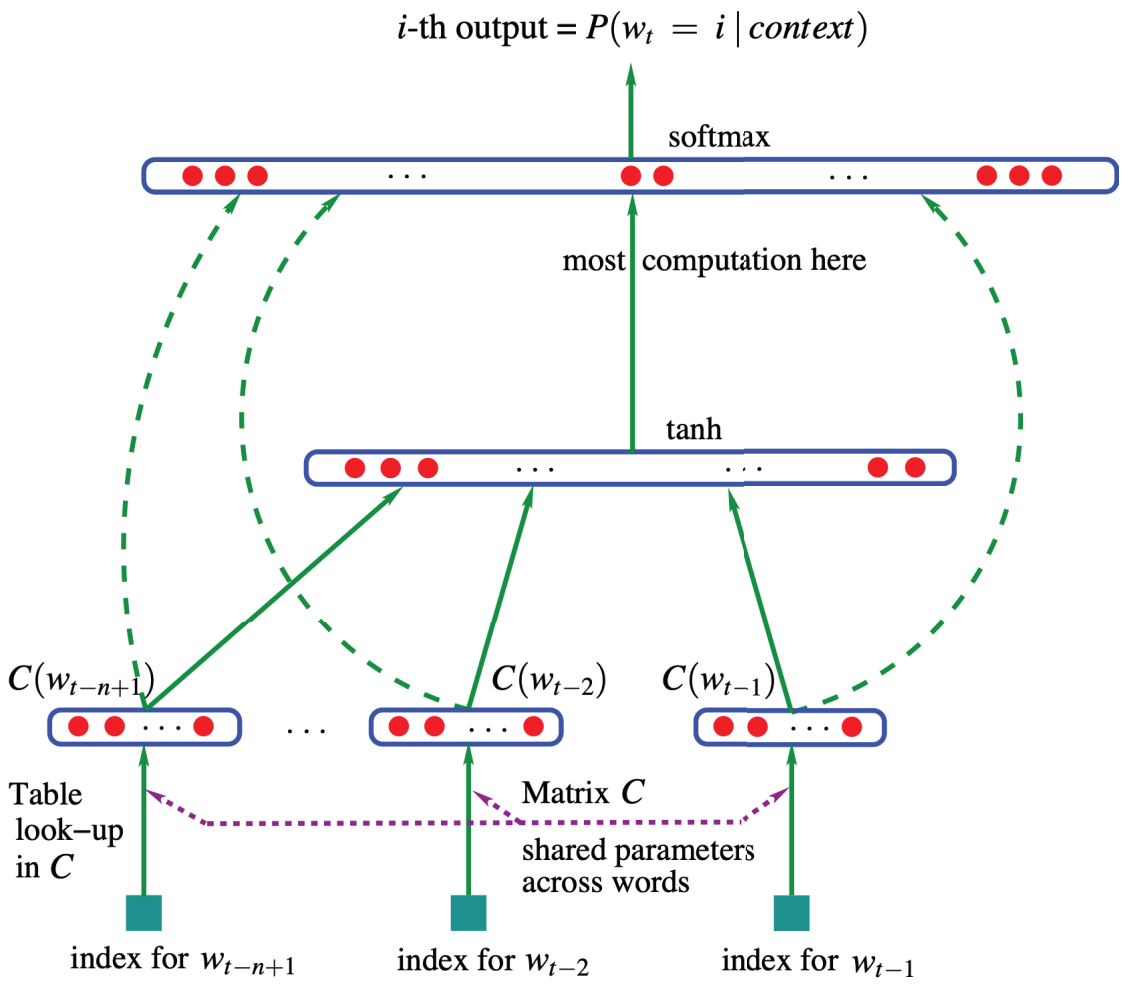


Figure 2.1. Probabilistic neural model. Figure borrowed from Bengio et at. [6], and highlights the two components f and g that form the language model. The first component converts words $w_{t-n-1}, ...w_{t-1}$ into vectors $C(w_{t-n-1}), ..., C(w_{t-1})$ using the lookup table C, the second component is a single layer perceptron function g which outputs probabilities of the next word w_t.

2.3 Recurrent Neural Networks

The problem with a n-context fully connected classifier g is that it cannot handle inputs of variable lengths unless we create the notion of a padding token for sequences shorter than length n or combining context windows for sequences longer than n. Words are sequences of characters, sentences are sequences of words, and paragraphs are sequences of sentences. Therefore, it is natural to utilize a classifier which leverages the sequential nature of textual data.

Using the marginal probability formulation of language modeling mentioned earlier,

a language model can calculate the probability of a sentence "It was a good day to play tennis," as $P(\text{It}) \times P(\text{was} \mid \text{It}) \times P(\text{a} \mid \text{It was}) ... \times P(\text{tennis} \mid \text{It was a good day to play})$.

A recurrent neural network (RNN) [12] keeps track of these probabilities at timestamp $t-1$ by maintaining a hidden state $\boldsymbol{h}_{t-1}$ which is a vector that is able to retain information from the prior inputs, in the case of language modeling $P(w_1, ...w_{t-1})$. Given a new input $C(w_t)$, it uses the hidden state to attain $P(w_1, ..., w_t)$.

Recurrent neural networks are composed of RNN cells that take two inputs, 1) the input at the current time-stamp $C(w_t) \rightarrow e_t$, and 2) the recurrent input from the previous timestamp $\boldsymbol{h}_{t-1}$, and cells that generate 2 outputs, 1) the output at the current cell o_t, and 2) the recurrent output at the current step h_t. The plain vanilla recurrent cell uses the following equations:

$$h_t = \sigma_h(W_h h_{t-1} + W_e e_t + b_1), \qquad \sigma_h \in \{ReLu, Tanh, Sigmoid\}$$
$$o_t = \sigma_o(W_o h_t + b_2) \qquad \sigma_o \in \{Softmax, Sigmoid\}$$

The learned parameters of the RNN are W_h, W_e and W_o

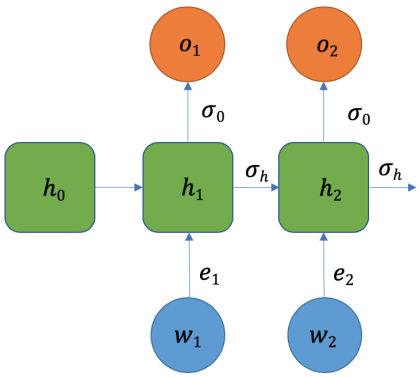


Figure 2.2. An recurrent neural network. At any timestamp t, input w_t (blue) is converted into e_t and fed into the recurrent cell (green). The recurrent unit uses two activation functions σ_o, σ_h to obtain h_t and o_t(orange). Usually only o_t is used for classification tasks. Usually the sigmoid or softmax activation functions are chosen for σ_o for classification choices, and tanh or sigmoid activation functions are chosen for σ_h to propagate information effectively.

2.4 Gated Recurrent Neural Networks

Recurrent neural networks rely on both recurrent and current inputs, so they must be optimized using the backpropagation through time algorithm [51]. Let $J_t(\theta; O_t, y_t)$ be the loss for the model given model parameters θ at timestamp t. We will represent this loss concisely as $J_t(\theta)$. Figure **2.3** highlights this process. Since W_h is shared across all timestamps,

$$\frac{\partial J_t}{\partial W_h} = \sum_{i=1}^{t} \frac{\partial h_i}{\partial W_h} \cdot \frac{\partial J_t}{\partial h_i} = \sum_{i=1}^{t} \left[\frac{\partial J_t}{\partial h_t} \cdot \frac{\partial h_i}{\partial W_h} \cdot \prod_{j=i}^{t-1} \frac{\partial h_{j+1}}{\partial h_j} \right]$$

Which means that just as the forward pass relies on recurrent inputs, backward propagation needs to go back through timestamps to be calculated. Figure **2.4** illustrates this.

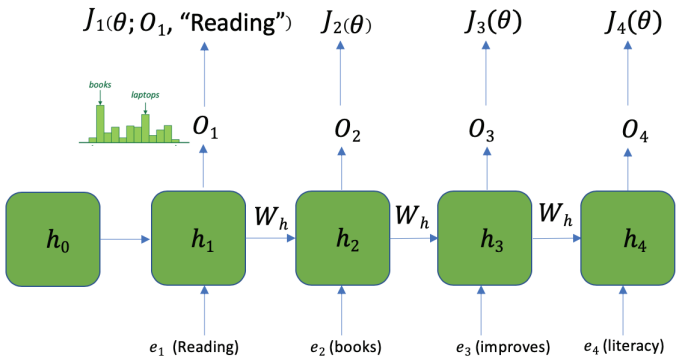


Figure 2.3. A recurrent neural network language model. At timestamp t, e_t is fed to the recurrent network to attain O_t, which is usually the output of a softmax activation function. Loss $J_t(\theta; O_t, y_t)$ is calculated and backpropagated at every timestamp.

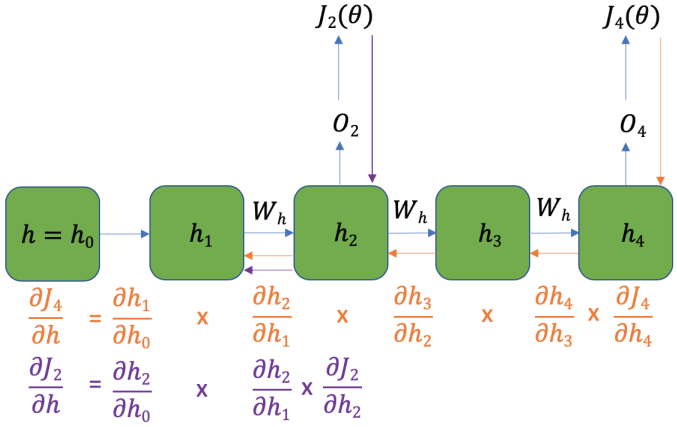


Figure 2.4. Backpropagation through time. We omit the details of $\frac{\partial J(\theta)}{\partial W_O}$ since it does not backpropagate through time.

A problem with simple recurrent neural networks is that the intermediary factors in

$$\prod_{j=i}^{t-1} \frac{\partial h_{j+1}}{\partial h_j}$$

tend to be a product of factors with magnitude less than 1 for activation functions such as sigmoid [**?**]. Repeated products of such terms will decay the gradient, hindering the ability for RNNs to optimize for NLP tasks [29]. As a result, it is difficult for gradients to be propagated from longer distances and learn long range input-output dependencies. In the example above, $\frac{\partial J_4(\theta)}{\partial h_0}$ requires four chain rule products through time, whereas $\frac{\partial J_1(\theta)}{\partial h_0}$ requires only two.

A *gated* recurrent neural network architecture, called long short-term memory (LSTM)[16] attempts to mitigate the vanishing gradients problem. The LSTM stores long-term information using additional memory, and providing additional functionality to this memory structure: the ability to 1) erase, 2) write and 3) read memory.

A gated neural network "gates" which sections of memory get erased/written/read using additional learned parameters. At a given timestamp t, an LSTM contains a hidden state h_t and cell state c_t, both of which are n-dimensional vectors. The gates of the LSTM are also n-dimensional, and their values lie in the range $(0, 1)$. The gate values are dynamic and are determined on the current context of the LSTM.

A newer gated network, GRU [8], proposes a simpler architecture which only has two gates, update, represented by z_t and reset, represented by r_t. While LSTMs have been more popular, GRU's have shown to yield similar performance, and there is no conclusive work that proves one consistently outperforms the other. A candidate state is calculated similarly to an RNN hidden state update, except the reset gate state r_t is used to determine how much information from h_{t-1} is used to determine $\tilde{h}_t$.

$$\tilde{h}_t = tanh(r_t \circ Uh_{t-1} + Wx_t) \qquad \text{Candidate state}$$

The GRU uses a linear interpolation between the previous state h_{t-1} and the current candidate state $\tilde{h}_t$ to calculate the new state.

$$h_t = (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1} \qquad \text{Hidden state}$$

The update gate value z_t indicates how much of prior hidden state h_{t-1} should be retained in the next state, and consequently how much information from $\tilde{h}_t$ is used to calculate current state.

$$z_t = \sigma(W_z x_t + U_Z h_{t-1}) \qquad \text{Update gate}$$

If r_t is a vector of ones, it is the exact same as an RNN hidden state update. If r_t is a vector of zeros, it forgets history completely.

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \qquad \text{Reset gate}$$

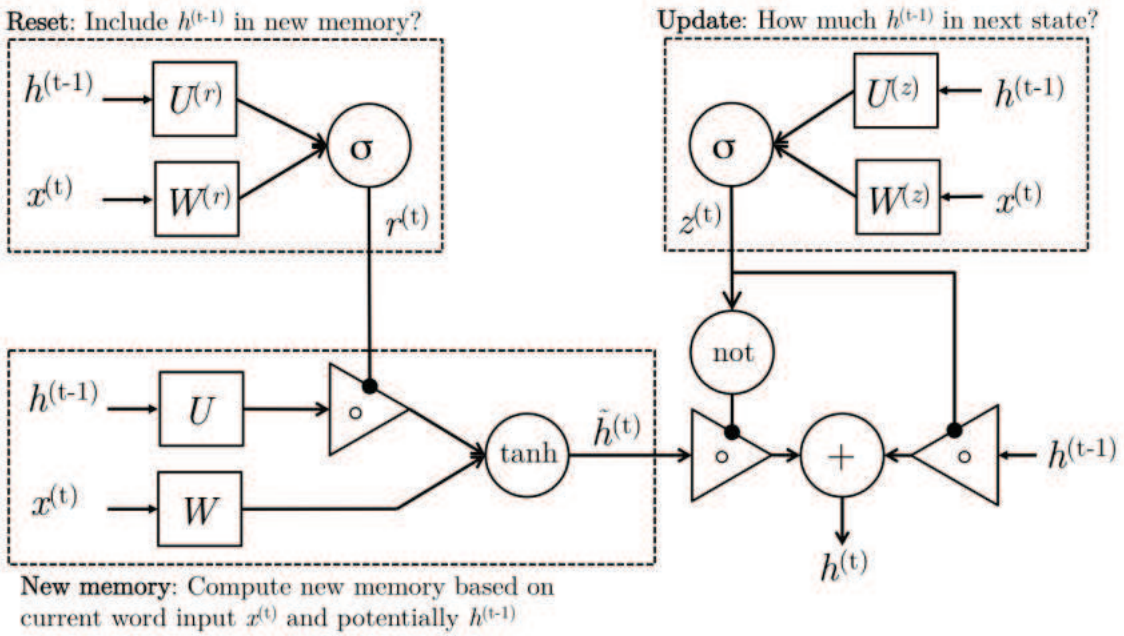


Figure 2.5. Detailed view of the GRU internals. Figure is borrowed from [26]. See text for further details and explanations.

The learned parameters of the GRU are W, U, W_r, U_r, W_z, U_z, which are roughly double that of simple RNNs.

2.5 Neural Attention

Language modeling is an unsupervised learning task in which at any given time the model has to predict the next word in a sentence given the past few. However, many NLP tasks, such as language generation, summarization and machine translation rely on sequential outputs, where the model is responsible for generating sequences as its output.

A Sequence-to-sequence model (usually abbreviated as seq2seq) [39] is an end-end neural network model which is made up of two recurrent neural network components: an encoder, which is responsible for encoding the input sequence $w_1, w_1, w_2 ... w_T$ of variable length T to a fixed-size context vector c_T, and a decoder, which can be thought of as a language model of the target sequence $P(y_1, ..., y_n | c_T)$.

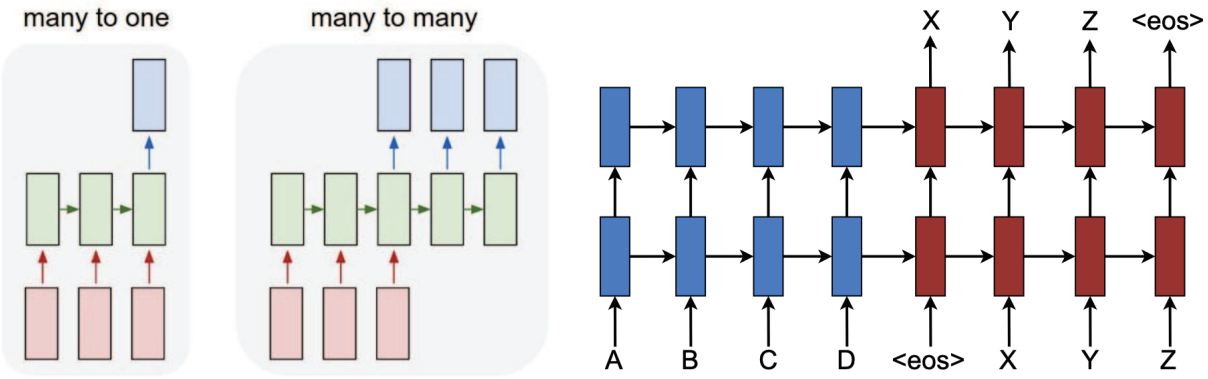


Figure 2.6. Sequence Modeling tasks. Left: Language modeling is considered a many–to-one sequence learning task, whereas tasks such as machine translation rely on many–to-many classification. Figure is borrowed from [13]. Right: The structure of a sequence–to-sequence classifier. The illustrated model has a unified encoder-decoder RNN, which means that the weights are shared between modules. Usually, a token representing the end of sequence `<eos>` signals the model to start the decoding phase of the task has started. This figure is borrowed from [25].

A problem with the seq2seq architecture is that it requires an entire sequence to be encoded into c_T before the decoder can start generating its output sequence [50]. This has led to degrading performance with longer input sequences, suggesting that the encoder creates a bottleneck that can be a source of problems for a sequence model [5]. Rather than utilizing a single context vector that is derived from encoder's last hidden state, the *attention mechanism* [5] creates shortcut connections between the decoder and the entire input sequence.

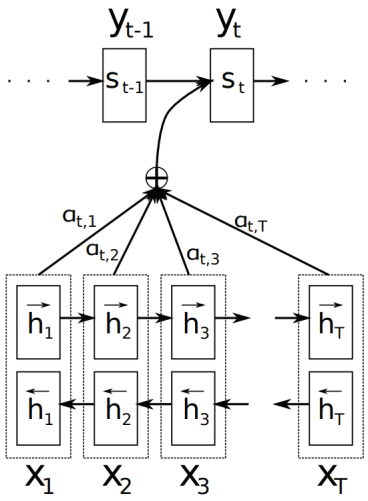


Figure 2.7. A seq2seq attention model. Figure borrowed from Bahadanu et al. [5]. This encoder relies on a Bidirectional-RNN, which consists of concatenating the hidden states of two RNNs, one going forwards, the other going backwards. The attention weights $\alpha_{t,u}$ indicate the amount of attention to pay to encoder output at timestamp u for generating the decoder output at timestamp t.

In a traditional seq2seq model, the encoder's final hidden state, $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_T}; \overleftarrow{\mathbf{h}_T}]$, is considered as $\mathbf{s}_0$, the decoder model's initial hidden state. With neural attention, weights $\boldsymbol{\alpha}$ are generated using a feed-forward network and provide an additional context vector $\mathbf{a}_t$ for the decoder.

Consider the task of translating a sequence $\mathbf{x}$ of length n into a sequence $\mathbf{y}$ of length m. The decoder's generated word at position $t \in 1, 2, ..., m$ is calculated using the information $\mathbf{s}_t = f(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{a}_T)$, where the attention vector $\mathbf{a}$ is a weighted sum of the encoder's hidden states.

$$\begin{aligned} \mathbf{a}_t &= \sum_{u=1}^{n} \alpha_{t,u} \mathbf{h}_i && \text{Attention vector} \\ \alpha_{t,u} &= softmax(score(\mathbf{s}_{t-1}, \mathbf{h}_{1...n}), u) && \text{Attention weights} \\ &= \frac{exp(score(\mathbf{s}_{t-1}, \mathbf{h}_u)}{\sum_{u'=1}^{n} exp(score(\mathbf{s}_{t-1}, \mathbf{h}_{u'})} \end{aligned}$$

The attention scores, also known as alignment scores, define how much of the source hidden state should be considered for each output. In the original work, the attention weights $\boldsymbol{\alpha}$ are calculated using a single hidden layer position-wise feed-forward-network, and is trained end-to-end with the rest of the model.

$$score(\mathbf{s}_t, \mathbf{h}_u) = \mathbf{v}_a \cdot tanh(\mathbf{W}_a[\mathbf{s}_t; \mathbf{h}_u]) \qquad \text{Attention score function}$$

$\mathbf{v}_a$ and $\mathbf{W}_a$ are the learnable parameters of the attention network.

In sequence classification, attention can be used to directly calculate the label of the sequence, without having to rely only on the encoder's hidden state.

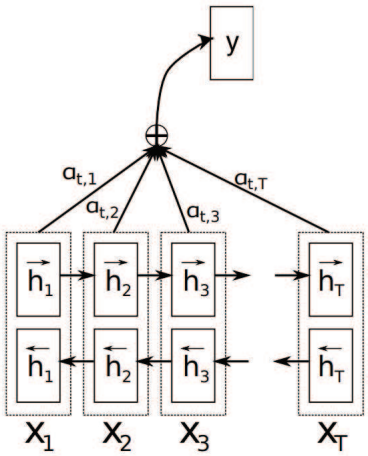


Figure 2.8. An attention-based RNN classification model. Whereas earlier models would rely on a combination of pooling over the hidden states and the final hidden state $\mathbf{h}_t$, an attention based model can learn a aggregating function which can be used to classify the sequence.

In summary, neural attention curbs some the problems associated with long distance relationships that traditional RNNs imposed. With neural attention, sequence classification possesses a fully connected component which can yield benefits over a simply recurrent network.

2.6 Transformer Neural Networks

The Transformer neural network [46] extends seq2seq by using an attention mechanism that now replaces recurrent units. This new mechanism, *self-attention*, increases the number of parameters and compute to process text but enables the transformer to behave similar to a fully connected network and leverage parallelism.

Self-attention is formulated as a function which maps a set of key-value $\{\mathbf{K}, \mathbf{V}\}$pairs to an output attention weight using a query $\mathbf{Q}$. The output value is a weighted sum

of the values, and each attention weight is calculated using a similarity function on the query and the respective key. Given the sequence length T and embedding dimension d, three weights $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ will perform a position-wise transformation to convert every embedding into its corresponding $\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t$

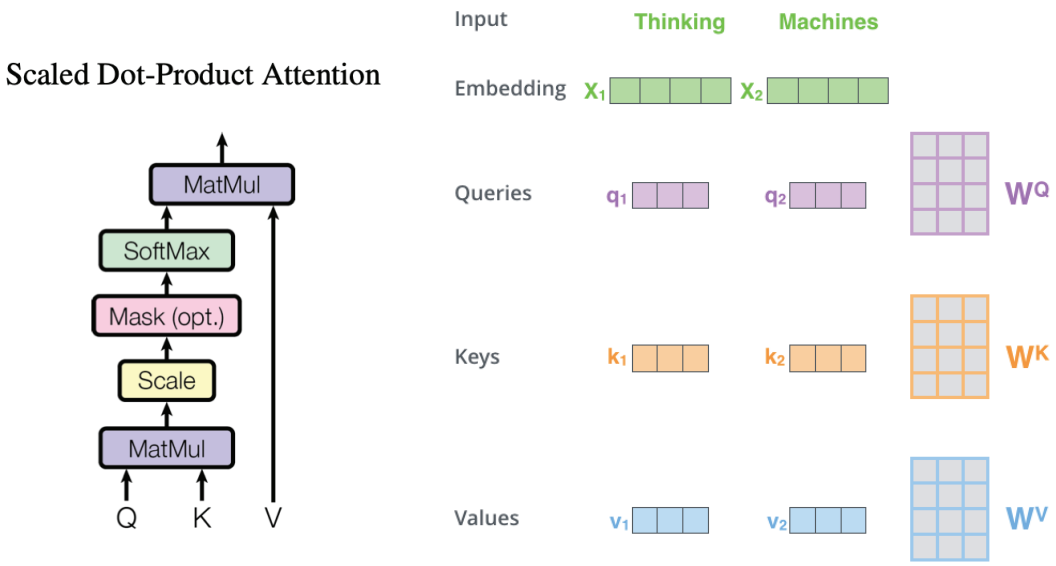


Figure 2.9. Left: QKV-self attention described in Vaswani et al. Right: How embeddings x_1, x_2 are converted to their respective queries keys and values. See texts for full details and descriptions. Figures borrowed from [46] and [1].

Given $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{d_k \times T}$, Self-attention is calculated with matrix multiplication, using scaled-dot product attention:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}$$

The weighted average softmax values $softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}) \in \mathbb{R}^{d_k \times d_k}$ are multiplied by $\mathbf{V} \in \mathbb{R}^{d_k \times T}$ to attain the self attended output embeddings for the next layer, which means that for most cases $d_k = d$. The input sequence $\mathbf{X} \in \mathbb{R}^{d \times T}$, and $Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{d \times T}$.

The transfomer architecture further utlizes self-attention by linearly projecting queries, keys and values h times with different projections such that $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{d \times T \times h}$. Each of the h "heads" conducts self-attention independently, and an additional weight $\mathbf{W}_O \in \mathbb{R}^{d \times h}$ projects these heads back into one dimension for the next layer.

$$MultiHeadAttention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(head_1, ...head_h)\mathbf{W}_O$$
$$head_i = Attention(\mathbf{Q}\mathbf{W}_Q^i, \mathbf{K}\mathbf{W}_K^i, \mathbf{V}\mathbf{W}_V^i)$$

Here the projections are parameter matrices $\mathbf{W}_Q^i, \mathbf{W}_K^i, \mathbf{W}_V^i \in \mathbb{R}^{d \times d_k}$. Since each of these heads can be calculated independently until the final linear transformation, multi-headed attention adds little overhead with proper parallelism and its ensemble properties can boost performance of the transformer network.

In a stacked transformer encoder, all keys, queries and values come from the output of the previous layer in the encoder. Each position has the flexibility to attend all positions in the previous layer. The transformer model contains no recurrence or convolution units, so there is no inherent way for the model to understand the order of tokens in the sequence. Therefore, the authors proposed adding positional encodings to input embedding vectors as means to inject the notion of order within these tokens.

2.7 Masked language models: BERT

Language models can be built up using context information from both the previous and the subsequent words in a sentence. *Bidirectional* RNNs can emulate this. However, the bottleneck of locality caused by RNNs provoked the usage of attention. Language modeling on a transformer model, however, is not straightforward, as self-attention makes is possible for a transformer embedding to ”cheat” by relying on self-attention to ”see itself,” something that could make the model trivially predict the target word in a multi-layered context. Unfortunately, it may not be sufficient to enforce a uni-directional self-attention mechanism, as subsequent context is crucial for language modeling. A recent language representation model titled BERT, short for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers [11], aims to train bidirectional representation from unlabeled text by jointly conditioning on both left and right context of all layers. The resulting model, *pre-trained* BERT, can then be fine-tuned with just an additional layer to cater to many NLP tasks. Upon its release, BERT achieved state of the art results on many tasks, such as SQUAD [32], GLUE [47], and MultiNLI [52].

BERT addresses the language modeling complications self-attention creates by using a *masked-language model* pre-training objective, which randomly masks tokens from the input. BERT’s objective is to predict the original emitted token of the masked word, relying

purely on its context. The final hidden vectors are fed into an output softmax over the vocabulary, a standard LM prediction task. 15% of all tokens in each sequence are masked at random. Of the 15% masked out tokens, 80% of the tokens are replaced with a special `[MASK]` token, 10% replaced with a random token from the vocabulary, and the token is untouched another 10% of the time.

In order to instill textual inference capabilities, BERT was also trained on a secondary task called *next sentence prediction* (NSP), which trains BERT to understand the relationship between two sentences. The NSP objective feeds an input example composed of two sentences, `[A; B]` and requires BERT to predict whether `A` actually precedes `B` or not. This dataset can be generated from any corpus with multiple sentences, and the sentence order is shuffled 50% of the time. Figure **2.10** gives an overview of pre-training and fine-tuning BERT.

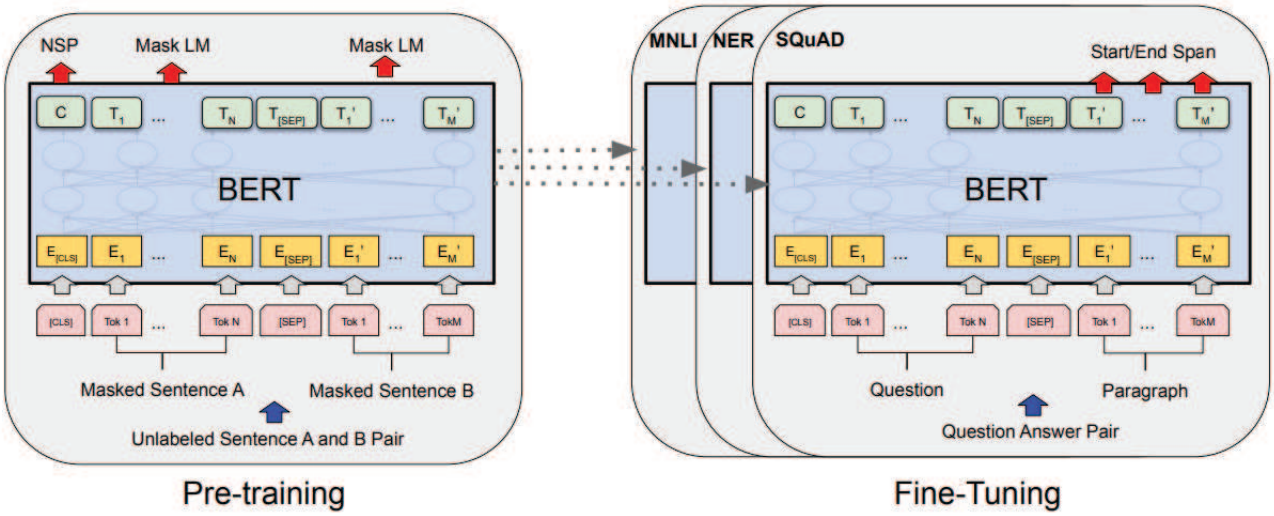


Figure 2.10. BERT model. Figure borrowed from [11]. Pre-trained BERT is trained using 1) Masked language modeling and 2) Next sentence prediction tasks on a large corpus. BERT can then be fine-tuned for other downstream NLP tasks. The `[CLS]` token is a special token added at the beginning of every input example, and is typically used for prediction tasks. `[SEP]` token is used for separating two sentences. This token separates `A` and `B` during pre-training NSP task, but can separate segments such as question-answer pairs during fine-tuning. Figure borrowed from original work.

Having introduced the limitations and the advancements in NLP models that led up to the transformer model, we now have the tools and modules for discussing the previous

state of the art models and the general purpose transformer-based language models we fine-tune as MISC classifiers.

CHAPTER 3

MOTIVATIONAL INTERVIEWING

3.1 Motivational Interviewing and Psychotherapy

Motivational Interviewing is a style of psychotherapy which focuses on motivating and encouraging a person's commitment to positive change. Counseling conversations can broadly lay anywhere in the spectrum of *directing* and *following*. In directing conversations, the counselor is usually deliberate and explicit about their instructions, advice, and information. In following conversations, the counselor refrains from injecting their opinions and advice and instead focuses on understanding and being interested in what the person has to say. Motivational Interviewing uses elements from both of these styles and relies on *guiding* the patient through listening and offering feedback and advice when necessary [27].

People who undergo motivational interviewing are often aware but ambivalent of the changes that they need to make to address their negative lifestyle patterns. Ambivalent people usually have arguments for both making the change and not, so they generally speak in a mixture of two kinds of talks, *change talk* and *sustain talk*. Change talk refers to the person's own reasons to make the change, while sustain talk refers to the person's reasons to sustain their current behavior. MI believes that couselors who take one side of the argument can often be met with resistance from the client who clings to the other side, so the objective of the counselor is to guide the client into voicing the reasons for change themselves. MI has shown to be particularly effective at treating patients who suffer from addictions and substance abuse problems, and has also been applied to natural language processing tasks [2].

The NLP task that most closely aligns with MI skill coding is **dialogue act** prediction. A dialogue act describes the function of an utterance in the scope of a conversation. These

help detect discourse structure and are pivotal for understanding dialogue [36]. Just like normal dialogue, there are benefits to being able to determine the function of an utterance in a Motivational Interviewing session. Motivational Interviewing Skill Code (MISC) was developed to help identify and code responses in MI, and is regularly utilized in training counsellors and to test and refine the principles of MI [34]. Natural language processing techniques have shown good results in coding MI sessions using machine learning techniques rather than human coders, and earlier work has explored the benefits of time and money that these models save [41]. Tables **3.1** and **3.2** show the MI codes used in this thesis. These codes are drawn from previous work in this domain [7] [53].

Code	Count	Description	Examples
FA	17468	Facilitate conversation	"Mm Hmm.", "OK.","Tell me more."
GI	15271	Give information or feedback.	"I'm Steve.", "Yes, alcohol is a depressant."
RES	6246	Simple reflection about the client's most recent utterance.	C: "I didn't smoke last week" T: "Cool, you avoided smoking last week."
REC	4651	Complex reflection based on a longer conversation or context.	C: "I didn't smoke last week." T: "You mean things begin to change".
QUC	5218	Closed question	"Did you smoke this week?"
QUO	4509	Open question	"Tell me more about your week."
	3869	Other MI adherent, e.g., affirmation, advising with permission, etc.	"You've accomplished a difficult task." "Is it OK if I suggested something?"
MIN	1019	MI non-adherent, e.g., confrontation, advising without permission..	"You hurt the baby's health for cigarettes?" "You ask them not to drink at your house."

Table 3.1. Therapist MISC labels and dataset used for the scope of this thesis. Table borrowed from Cao et al. [7]

Code	Count	Description	Examples
FN	47715	Follow/ Neutral: unrelated to changing or sustaining behavior.	"You know, I didn't smoke for a while." "I have smoked for forty years now."
CHANGE	5099	Changing unhealthy behavior.	"I want to stop smoking."
SUSTAIN	4378	Sustaining unhealthy behavior.	"I really don't think I smoke too much."

Table 3.2. Client MISC labels and dataset used for the scope of this thesis. Table borrowed from Cao et al. [7]

3.2 Natural language processing for MISC

In this section we outline the advancements made in predicting MISC labels in Cao et al. [7], which serves as a precursor and baseline to this thesis. Given an annotated corpus of MISC labels, the authors formulated two tasks: 1) Categorization and 2) Forecasting. The goal of the categorization task is to classify the MISC of an *anchor* utterance u_n given a snapshot of the conversation: $u_1, u_2 ... u_n$, and the identities $s_1, s_2 ... s_n$. The following subsections cover the paper's approaches and highlight the state-of-the-art results.

3.2.1 MISC Classification Setup

The MISC dataset we use is composed of psychotherapy sessions collected and labeled for motivational interviewing dissemination studies. Encounters include hospital settings [33], outpatient clinics [4] and college alcohol interventions [20] [28] [45]. All sessions were annotated with MISC codes [2]. Following Cao et al. [7], we split the data into 243 MI sessions for training, 110 for testing, and 24 for development. We follow the labels formulated in Xiao et al. [54] , which adopts 3 client codes {`CHANGE, SUSTAIN, FN`} and 8 therapist codes {`FA, RES, REC, GI, QUO, QUC, other MIA, MIN`}. Refer to Table **3.1** and **3.2** for more details on these labels.

3.2.2 Encoding Dialogue

Since categorization is a classification task given some dialogue history, the goal is to convert the sequence of utterences into a fixed size vector that will be used for classification at prediction time. The authors used a hierarchical GRU encoder (HGRU) [21] to encode dialogues using a two level, utterance and dialogue, encoder. The utterance is encoded bidirectionally and the i-th utterance is known as v_i. The dialogue encoder is an unidirectional GRU that operates on a concatenation of utterance vectors v_i and a trainable vector representing the speaker s_i. The final hidden state of the dialogue GRU aggregates the entire dialogue history into a vector H_n.

3.2.3 Word-level Attention

To give this hierarchical recurrent models the ability to attend at a word level, the authors propose a couple attention mechanisms for the ability for the kth word in anchor utterance, v_{nk} to attend to the jth word in ith utterance, v_{ij}. The attention weighted vector

$\boldsymbol{a_{ij}} = \sum_k \alpha_j^k \boldsymbol{v_{nk}}$, where

$$\alpha_j^k = \frac{exp(f_m(\boldsymbol{v_{nk}}, \boldsymbol{v_{ij}}))}{\sum_{j'} exp(f_m(\boldsymbol{v_{nk}}, \boldsymbol{v_{ij'}}))}$$

A combining function f_c combines original v_{ij} and the weighted word vector $\boldsymbol{a_{ij}}$ into a new representation $\mathbf{z_{ij}}$

$$\boldsymbol{z_{ij}} = f_c(\boldsymbol{v_{ij}}, \boldsymbol{a_{ij}})$$

Method	f_m	f_c
BiDAF [35]	$\boldsymbol{v_{nk}} \boldsymbol{v_{ij}^T}$	$[\boldsymbol{v_{ij}};\ \boldsymbol{a_{ij}};\ \boldsymbol{v_{ij}} \odot \boldsymbol{a_{ij}};\ \boldsymbol{v_{ij}} \odot \boldsymbol{a'}]$
GMGRU [48]	$\boldsymbol{w^e} \tanh(\boldsymbol{W^k v_{nk}} + \boldsymbol{W^q}[\boldsymbol{v_{ij}}; \boldsymbol{h_{j-1}}])$	$[\boldsymbol{v_{ij}}; \boldsymbol{a_{ij}}]$

Table 3.3. The word-level attention mechanisms used in Cao et al. [7].

3.2.4 Utterance-level Attention

The relevancy of an utterance to anchor utterance's label is modeled using a multi-headed, single layered transformer network. Two different variants of transformer-attention mechanisms are compared: *anchor-based* and *self* attention. The anchor-based attention uses $\mathbf{Q} = \mathbf{v}_n$ and $\mathbf{K} = \mathbf{V} = [\mathbf{v}_1 ... \mathbf{v}_n]$ while self-attention sets all three matrices to $[\mathbf{v}_1 ... \mathbf{v}_n]$.

3.2.5 Label Imbalance

Each classifier is evaluated on its performance based on the F1 score it achieves on the individual MISC labels. Since it is crucial to detect rarer MISC labels, such as `MIN` and `SUSTAIN`, overal unweighted F1-score, also known as the macro F1-score is also reported. For each model, the best checkpoint on the development set is used to calculate the F1-scores on the test set. In order to address the label imbalance present in the dataset, a focal loss [22] optimization function is adopted. For a label l with a model softmax probability of p_t, the loss is defined as

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma log(p_t)$$

The paper uses α as $\{1.0, 1.0, 0.25\}$ for `CT`,`ST` and `FN` respectively, and $\{0.5, 1.0, 1.0, 1.0, 0.75, 0.75, 1.0, 1.0\}$ for `FA, RES, REC, GI, QUC, QUO, MIA, MIN`, and fixing $\gamma = 1$, which is effectively just adding class weights to the cross entropy loss function. We do not use any strategy to address label imbalance in our experiments.

3.2.6 Results

The best configuration for modeling a patient observer does not rely on any word or utterance attention, which most likely means that client responses do not require much context to categorize. The best therapist observer model uses `GMGRU` word attention and `ANCHOR` utterance attention.

Method	macro	FN	CHANGE	SUSTAIN
Majority	30.6	**91.7**	0.0	0.0
$\text{BiGRU}_{\text{ELMo}}$	52.9	87.6	**39.2**	32.0
CONCAT	51.8	86.5	38.8	30.2
GMGRU	52.6	89.5	37.1	31.1
BiDAF	50.4	87.6	36.5	27.1
$\mathcal{C}_C$	**53.9**	89.6	39.1	**33.1**

Table 3.4. RNN based MISC classification results on client codes. $\mathcal{C}_C$ is a simple $\text{MLP}(H_n) + \text{MLP}(v_n)$ model which does not rely on any attention mechanism.

Method	macro	FA	RES	REC	GI	QUC	QUO	MIA	MIN
Majority	5.87	47.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\text{BiGRU}_{\text{generic}}$	60.2	94.5	50.5	49.3	72.0	70.7	80.1	54.0	10.8
$\text{BiGRU}_{\text{ELMo}}$	62.6	94.5	51.6	49.4	70.7	72.1	80.8	57.2	24.2
CONCAT	61.0	94.5	54.6	34.3	73.3	73.6	81.4	54.6	22.0
GMGRU	64.9	94.9	**56.0**	54.4	**75.5**	**75.7**	**83.0**	**58.2**	21.8
BiDAF	63.8	94.7	55.9	49.7	75.4	73.8	80.7	56.2	24.0
$\mathcal{C}_T$	**65.4**	**95.0**	55.7	**54.9**	74.2	74.8	82.6	56.6	**29.7**

Table 3.5. RNN based MISC classification results on therapist codes. Models use the same configuration as Table 3.2, but tuned for therapist anchor utterance codes.

In both tables, the first set utilizes only the anchor utterance, while the second sets of baselines use the context in some manner as well. `CONCAT` is a simple BiGRU model which utilizes contextualized `ELMo` embeddings. `generic` embeddings utilize `GloVe` vectors. The best client model $\mathcal{C}_C$ suggests that a client's MISC code is predominantly dictated by their utterance and not heavily reliant on the conversation history. However, a simple $\text{MLP}(H_n)$ module seems to boost the performance of the model, suggesting a summarization of the context is sufficient for prediction.

Model $\mathcal{C}_T$ relies on both GMGRU-based word-level attention and anchor-based multi-head sentence-level attention, suggesting that therapist MISC codes rely a lot more on the context of the conversation. Anchor-based attention also emphasizes the importance of the current utterance while considering previous statements. The complexity of this model indicates that more complex architectures like the transformer models may perform even better on such a task.

CHAPTER 4

APPROACH: REPRESENTING DIALOGUE TO TRANSFORMERS

We begin this chapter by setting up our experiments, elaborating on our training schema and interpretations studies.

4.1 Dialogue Encoding

Our training dataset consists of roughly $60,000$ annotated client and $50,000$ therapist utterances. Since transformer networks process data without explicit recurrence, every example's context will need to be joined, concatenated and fed to the model with the example's utterance. As such we consider a few different methods to inject the notion of dialogue to these models.

We start by establishing two baseline models which predict the MISC labels of an utterance with no context, and call these $\texttt{UTTER}_C$ and $\texttt{UTTER}_T$. Since $\mathcal{C}_T$ did not rely heavily on context and $\texttt{UTTER}_C$ relies on no context, it should be possible for this simple $\texttt{UTTER}_C$ to achieve state-of-the art performance on client MISC.

Following [7] we rely on context history of at most 8 utterances for all context-based classifiers. We establish baseline models $\texttt{CONCAT}_C$ and $\texttt{CONCAT}_T$, which simply join utterances as different sentences. No special `[SEP]` token is used to separate these sentences to the model, and the identity of the speaker is not indicated to the model.

We then train two models $\texttt{SEP}_C$ and $\texttt{SEP}_T$, which chunk the input with appropriate separators for transformer models. The usage of separators should greatly increase the performance of the model, as this input format follows closely with the usage of the these symbols in the pre-training phase of the transformer models. To measure the effectiveness of speaker identity, we add the identity of each utterance speaker as a cue to the models, generating $\texttt{SPEAKER}_C$ and $\texttt{SPEAKER}_T$. We test different variants of encoding the speaker in the dialogue. $\texttt{SPEAKER}_C$ and $\texttt{SPEAKER}_T$ rely on prepending the speaker identity to every

utterance as natural language. SPEAKER-SYM$_C$ and SPEAKER-SYM$_T$ introduce psychotherpy specific special tokens [U], [P], [T] to indicate an utterance, patient speaker, and therapist speaker to the model. SPEAKER-SYM-SEP$_C$ and SPEAKER-SYM-SEP$_T$ variants of the models rely on initializing these seperators to the pre-trained model's [SEP] token. Finally, we train SPEAKER-SPAN$_C$ and SPEAKER-SPAN$_T$ which rely on special tokens [U], [/U], [P], [/P], [T], [/T] to indicate spans of the texts. A span representation approximately doubles the number of special tokens used in the text, but may yield benefits with adding more structure to the text. As we did before, we train variants SPEAKER-SPAN-SEP$_C$ and SPEAKER-SPAN-SEP$_T$, which are fine-tuned from the embedding vector of [SEP].

We also explore training *unified* MISC classifiers, where one model is trained on both therapist and patient data. These models will be indicated with the same names as above but will lack the subscript indicating the agent they are trained to classify.

Method	BERT encoding
UTTER	[CLS] u_t [SEP]
CONCAT	[CLS] $[u_1; ...; u_t]$ [SEP]
SEP	[CLS] u_1 [SEP] ... [SEP] u_t] [SEP]
SPEAKER	[CLS] T: u_1 [SEP] ... [SEP] C: u_t [SEP]
SPEAKER-SYM	[CLS] [T] u_1 [SEP] [C] ... [SEP] [C] [U] u_t [SEP]
SPEAKER-SPAN	[CLS] [T] u_1 [/T] [SEP] [C] ... [/T] [SEP] [C] [U] u_t [/U] [/C] [SEP]

Table 4.1. Summary of dialogue data encodings we use in this work. RoBERTa uses <s> and </s> instead of [CLS] and [SEP]. It also uses two separator tokens between utterances in place of only one that BERT uses.

4.2 Training and Optimization

We use the HuggingFace transformers library [1] to retrieve our pre-trained transformer models and word-piece tokenizers. For our experiments we use 3 models.

- bert-base-cased, which is the standard BERT [11] model with case-sensitive embeddings. BERT is trained using two special tokens [CLS] and [SEP]. For our train-

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

ing purposes separate every utterance with `[SEP]` and append it to the end of every input to serve as our end of text token.

- `bert-base-cased-conversational`, which is a BERT model trained on dialogue corpra from the internet and media such as movies and books[2]. Since this model is trained identically to BERT, we will use the same input patterns.

- `roberta-base`, a BERT-like model which was trained using more data, larger batches and longer time [23]. RoBERTa relies on `<s>` and `</s>` tokens instead of `[CLS]` and `[SEP]`, and separates every pair of documents with two `</s>`, first one serving as the end of text for the previous sentence, the latter indicating the start of the next sentence. Each sequence ends with only one trailing `</s>`, marking the end of the last document to the model.

All three models use and output 768 dimensional embeddings. For classification, BERT based models use two fully connected layer networks that are fed the final representation of the `[CLS]` or the `<s>` embedding. For the BERT-based models, this means a

$$[768 \rightarrow 768 \rightarrow 3 \text{ (patient) or } 8 \text{ (therapist)}$$

classification head. RoBERTa adds an additional layer, utilizing a slightly larger head:

$$[768 \rightarrow 768 \rightarrow 768 \rightarrow 3 \text{ (patient) or } 8 \text{ (therapist)}]$$

For unified models, we add a 3+8=11 logit classification head, and loss is only calculated using the logits that correspond to the anchor utterance's speaker.

Due to the computational requirements of training transformer models, we conducted preliminary studies on hyperparmeter values. We fix our experiments to use reasonable or best performing parameters with the compute costs in consideration. Each model is trained for 10 epochs, using two NVIDIA Titan X GPUs. We use an effective batch size of 128 using gradient accumulation, and checkpoint 4 models spread evenly across training time. We use the AdamW [24] optimizer with a learning rate of 3×10^{-5}. We use linear learning rate warmup schedules, with a warmup period of 1000 warmup steps for BERT

[2]https://github.com/deepmipt/deeppavlov

and conversational-BERT models, and the first 6% of training time for RoBERTa models following suggestions given in their respective publications [11] [23]. Learning rate is set to 3×10^{-5}, and we use a weight decay of 0.1. We work with context length of at most eight, and every training batch's length is adaptively generated to match the length of the longest example in the batch. We use the standard cross entropy loss, except when training a unified model, for which we mask the logits of the non-anchor speaker's MISC codes by setting them to -10000 before we calculate softmax probabilities $\hat{\mathbf{p}}_t \in \mathbb{R}^j$.

$$loss(\hat{\mathbf{p}}_t, y_t) = -p_{t,y_t} + log(\sum_j exp(p_{t,j}))$$

4.3 Model Interpretation techniques

Dissecting the attention weights of a model has been an effective interpretation tool for seq2seq RNN-based models [5][15]. However, transformer-based self-attention has been much more difficult to visualize in the same manner because of the number of heads, layers and the higher-order inter-layer dependencies present within self-attention models.

We explore the feasibility of interpreting these models using gradient-based saliency map approaches [3], and evaluating the reliability of the method using input reduction techniques mentioned in [14]. Our visualization interface is powered by AllenNLP's framework[3], on which we bootstrap the usage of dialogue data.

Finally we analyze self-attention weights across the models, their layers and heads, which end up being 12 layers $\times$12 heads = 144 attention maps for all three of our models. While it is unlikely that we can extract the role of a specific self-attention map as it influences the prediction, it would aid in understanding model tendencies and behaviors. To achieve this, we build upon the exBERT visualization [17][4] with the ability to process dialogue information.

[3]https://github.com/allenai/allennlp-demo

[4]https://exbert.net/

CHAPTER 5

RESULTS

Since we run seven experiments across three different models, we will only report the best performing model for each task. For each experiment, the checkpoint with the best dev performance is used to evaluate test-set performance. We report the final test-time results and compare against the existing state-of-the-art baselines $\mathcal{C}_C$ and $\mathcal{C}_T$. Our evaluation methods are the same as section **3.2.6**

5.1 Client Predictions

Method	Model	macro	FN	CHANGE	SUSTAIN
Majority	-	30.6	**91.7**	0.0	0.0
$\mathcal{C}_C$	MLP	53.9	89.6	39.1	**33.1**
$\texttt{UTTER}_C$	RoBERTa	57.7	90.9	39.6	42.6
$\texttt{CONCAT}_C$	BERT	52.3	90.4	35.4	31.2
$\texttt{SEP}_C$	Conversational-BERT	56.3	90.5	38.3	40.1
$\texttt{SPEAKER}_C$	Conversational-BERT	56.7	89.4	38.7	42.0
$\texttt{SPEAKER-SYM}_C$	RoBERTa	**53.9**	89.6	39.1	42.1
$\texttt{SPEAKER-SYM-SEP}_C$	RoBERTa	58.9	89.8	42.7	44.3
$\texttt{SPEAKER-SPAN}_C$	Conversational-BERT	61.2	90.8	45.2	47.6
$\texttt{SPEAKER-SPAN-SEP}_C$	RoBERTa	**61.5**	90.8	**45.3**	**48.5**
$\Delta = \texttt{SPEAKER-SPAN-SEP}_C - \mathcal{C}_C$	-	+7.6	-0.9	+3.9	+3.8

Table 5.1. Transformer based MISC classification results on client codes.

Table **5.1** summarizes client prediction results. Early experiments show that three models trained under the $\texttt{UTTER}_C$ outperform $\mathcal{C}_C$ and achieve state-of-the-art performances without any need for context. Interestingly, all context-based transformer models, starting from $\texttt{CONCAT}_C$ until $\texttt{SPEAKER-SPAN}_C$ are outperformed by the baseline in at least one of the labels. This suggests certain context encodings may behave like noise to the model and complicate the task. The difficulty in matching the $\texttt{UTTER}_C$ performance in most

context-based inputs sheds light that client MISC codes may not be as contextual as therapist codes. Since it only outperforms other models in only one setting, we can see that BERT is consistently outperformed by RoBERTa and Conversational-BERT for this task, suggesting that the latter two models are better equipped for dialogues, since they are trained on multi-document texts, whereas BERT is only trained on a max of two sentences per example for NSP task.

5.2 Therapist Predictions

Table **5.2** summarizes therapist prediction results. These results indicate that context is more important to predict therapist MISC than client MISC, as every contextual encoding starting from $\texttt{SPEAKER-SYM}_T$ outperforms $\texttt{UTTER}_T$ encoding. With proper encoded dialogue, we are able to achieve state-of-the-art in every category. Conversational BERT consistently outperforms RoBERTa for categorizing the therapist codes, most likely since it was trained on dialogue data. The therapist MISC categorization results suggest that client MISC falls into either too broad of a category, or relies less on the history of the conversation than classifying therapist MISC.

Method	Model	macro	FA	RES	REC	GI	QUC	QUO	MIA	MIN
Majority	-	5.87	47.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\mathcal{C}_T$	HGRU + Anchor	**65.4**	**95.0**	55.7	**54.9**	74.2	74.8	82.6	56.6	**29.7**
$\texttt{UTTER}_T$	RoBERTa	65.01	94.7	54.4	51.3	75.0	75.3	82.6	59.1	27.7
$\texttt{CONCAT}_T$	RoBERTa	59.68	94.0	51.4	50.4	73.5	73.8	58.8	51.9	23.7
$\texttt{SEP}_T$	BERT	65.70	94.3	54.3	54.4	75.1	74.3	79.6	58.5	24.9
$\texttt{SPEAKER}_T$	Conv-BERT	65.81	94.5	56.5	55.0	75.9	75.5	80.6	58.6	29.9
$\texttt{SPEAKER-SYM}_T$	Conv-BERT	66.18	94.7	57.2	55.0	76.7	75.9	81.2	58.7	30.1
$\texttt{SPEAKER-SYM-SEP}_T$	Conv-BERT	66.31	94.7	57.4	54.9	76.8	76.1	81.5	58.7	30.1
$\texttt{SPEAKER-SPAN}_T$	Conv-BERT	67.1	95.0	58.1	55.6	77.7	76.9	82.8	59.3	31.4
$\texttt{SPEAKER-SPAN-SEP}_T$	Conv-BERT	**67.53**	**95.0**	**58.7**	**56.3**	**78.0**	**77.3**	**83.1**	**59.7**	**32.2**
$\Delta = \underline{\text{score}} - \mathcal{C}_T$	-	+2.13	+3.0	+1.4	+3.4	+3.8	+2.5	+0.5	+3.1	+2.5

Table 5.2. Transformer based MISC classification results on therapist codes.

5.3 Unified Predictions

Method	FN	CHANGE	SUSTAIN
SPEAKER-SPAN-SEP$_C$	90.8	**45.3**	**48.5**
SPEAKER-SYM-SEP	91.3	47.1	48.0
SPEAKER-SPAN-SEP	**92.0**	**48.5**	**50.6**
$\Delta = \underline{\text{unified}} - \underline{\text{separate}}$	+0.7	+3.2	+3.1

Table 5.3. Unified Transformer based MISC classification results on patient codes.

Method	FA	RES	REC	GI	QUC	QUO	MIA	MIN
SPEAKER-SPAN-SEP$_T$	**95.0**	**58.7**	**56.3**	**78.0**	**77.3**	**83.1**	**59.7**	**32.2**
SPEAKER-SYM-SEP	95.0	58.3	54.2	77.5	78.1	82.6	50.1	31.8
SPEAKER-SPAN-SEP	95.0	58.3	56.1	**78.8**	**77.9**	**83.5**	**59.7**	**32.0**
$\Delta = \underline{\text{unified}} - \underline{\text{separate}}$	+0	-0.4	-0.2	+0.8	+0.6	+0.4	+0	-0.2

Table 5.4. Unified Transformer based MISC classification results on therapist codes.

Table **5.3** summarizes unified prediction results. The unified approach improves all client MISC code predictions. Perhaps the model is able to understand dialogue and context well enough to generalize it to client classification codes, revealing some structure of the context which training simply on client MISC could not uncover. We observe a general increase in performance suggesting that MISC prediction can be leveraged as a unified modeling problem, and that both the agents do share some common representation of psychotherapy dialogue. We release the code and runtime scripts used to run these experiments. [1]

5.4 Saliency Maps

In order to probe what tokens are the most influential towards making a prediction, we visualized the magnitude of gradients with respect to the input embeddings with respect to a label of our choosing. Some anchor utterances, such as *Okay* are strong indicators

[1] https://github.com/utahnlp/bert-therapy

of FA MISC. However, our findings show that the largest input gradients are made to transformer special tokens. The magnitude of these gradients are four or more times larger than for tokens, and obfuscate the meaning of token gradients.

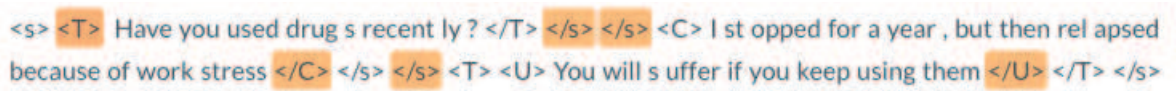


Figure 5.1. Saliency map interpretation of a sample utterance on RoBERTa SPEAKER-SPAN-SEP. The largest 6 gradients correspond to special tokens, which played an aggregating role during their pre-training and fine-tuning phase.

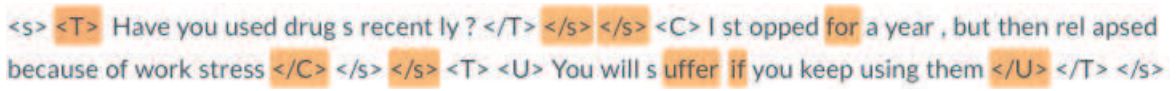


Figure 5.2. More gradients for the saliency map interpretation of a sample utterance on RoBERTa SPEAKER-SPAN-SEP. The magnitude of token gradients is cryptic and not easily explicable. In this MIN example, the word suffer is intuitively a big indicator of the MISC code, but is the eigth most crucial token for the model.

It was pointed out in [14] that non-linear, higher order relationships between the tokens make deep RNNs difficult to probe. Since transformers incorporate self-attention, which means every token is connected to every other token with just one hop, this phenomenon seems to be amplified, making the saliency map based interpretation uninterpretable. It is worth noting that all three models used for experiments stack 12 transformer layers, so further studies should explore if the gradients are better explained at any higher levels.

5.5 Attention Maps

Self-attention does not enforce any distance-related constraints on tokens. It loosely embeds the notion of distance through position its encodings, but it's not clear how these positions affect self-attention. Since dialogue is lengthier than traditional NLP tasks, it may take advantage of self-attention to capture long-range dependencies. However, our analysis has found that self-attention offers too much freedom to be interpretable through individual attention maps.

There are some themes present in these attention maps. Many attention heads tend to attend to the previous or next token, something which indicates that the models are aware of token positions. Few of the attention heads seem to attend between different parts of speech, but these heads usually also attend on seemingly random tokens.

Self-attention becomes less interpretable as we examine deeper layers, suggesting that self-attention leverages on higher-order relationships between tokens in order to understand the complexity of language.

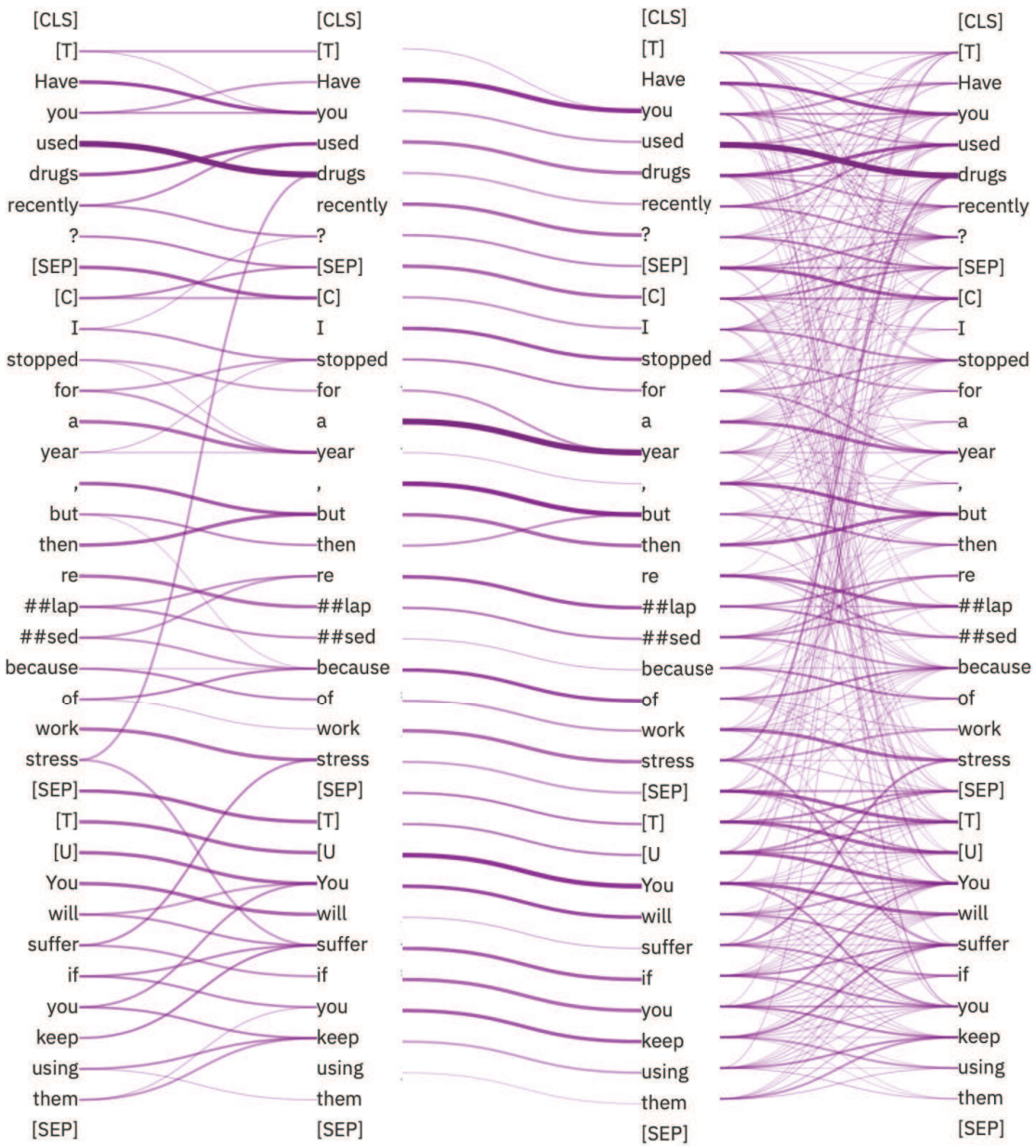


Figure 5.3. Examples of attention maps from analysis on psychotherapy dialogue data. Only the top 10% attenion connections are shown. Left: typical attention maps in the first layer, shows no real trends. Middle: An attention head responsible for making tokens attend to the next token, with a couple exceptions. It is also from the first layer. Right: An attention head at layer 12, at which point it is difficult to discern the functionality of the attention head.

CHAPTER 6

CONCLUSIONS

The performance of pre-trained transformers outpaces RNN-based classifiers on MISC performance. Given dialogue in the right format, transformer models can leverage self-attention to optimize MISC categorization effectively compared to earlier RNN models which experimented with multiple attention mechanisms between utterances and words across those utterances. Interestingly, a concatenating the context and the anchor utterance together could not match the performance of just training on the anchor utterance, suggesting that the transformer position encodings are not powerful enough to bias more towards the word embeddings at the end of the sentence.

Adding special tokens boosted performance greatly, especially when initialized to a separator token. From the experiments on the client MISC codes, it seems that RoBERTa may be able to initialize and fine-tune these special tokens better than the BERT models. However, Conversational-BERT's pre-training data gave it the edge to understand dialogue better and perform consistently better in the therapist experiments.

Even though adding span tokens can double the number of artificial tokens to the model, it seems to perform much better than adding a prefix symbol. Since transformer models are pre-trained on spans of text, it suggests that transformer models can specialize at learning between two special tokens, and results indicate that they are effective learning across these spans as well.

Gradient-based interpretation approaches do not yield interpretable results, most likely because self-attention provides shortcut connections in the model and promotes higher-order relationships between the embeddings, especially at deeper layers. This explains the semantic knowledge that transformer models possess, but at the expense of making it harder to interpret these transformer models. Attention weight interpretation attempts result in the similar conclusions: it is hard to pinpoint the role of a specific attention head

at a specific layer. However, these visualizations did indicate that transformer models can indeed understand position encodings and that self-attention requires many heads and layers to understand language properly. Overall, our studies show that off-the-shelf interpretation mechanisms are lacking the ability to sufficiently probe and explain transformer models. Future work should address this and develop more robust techniques as lack of model insights hinder our ability to deploy these models into real-world therapy sessions otherwise.

In this thesis, we were able to test transformer models' ability to generalize from language modeling to a dialogue task, particularly predicting the MISC code of an utterance given some dialogue. Given that these models were fine-tuned on significantly less steps and at a much smaller learning rate, transformer-based models may be robust alternatives to hierarchical RNN-based models. We show that MISC prediction models do not have to be client or therapist specific, and that unified learning objectives can improve state-of-the-art results without relying on modifying or complicating cross-entropy loss minimization objective. Our results also indicate that the client MISC may need to be expanded from just three labels since those codes only weakly rely on the context of the conversation, something which can further the development and adaptation of MISC labels.

REFERENCES

[1] Alamar, J. The illustrated transformer. *jalammar.github.io/* (2018).

[2] Atkins, D. C., Steyvers, M., Imel, Z. E., and Smyth, P. Scaling up the evaluation of psychotherapy: evaluating motivational interviewing fidelity via statistical text classification. *Implementation Science 9*, 1 (2014), 49.

[3] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Mueller, K.-R. How to explain individual classification decisions. *Journal of Machine Learning Research* (2009).

[4] Baer, J. S., Wells, E. A., Rosengren, D. B., Hartzler, B., Beadnell, B., and Dunn, C. Agency context and tailored training in technology transfer: A pilot evaluation of motivational interviewing training for community counselors. *Journal of substance abuse treatment 37*, 2 (2009), 191–202.

[5] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations* (2014).

[6] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. A neural probabilistic language model. *Journal of machine learning research 3*, Feb (2003), 1137–1155.

[7] Cao, J., Tanana, M., Imel, Z., Poitras, E., Atkins, D., and Srikumar, V. Observing dialogue in therapy: Categorizing and forecasting behavioral codes. In *Proceedings of ACL 2019* (2019).

[8] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[9] Colby, K. M. *Artificial Paranoia: A Computer Simulation of Paranoid Process*. Pergamon Press, 1975.

[10] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res. 999888* (Nov. 2011), 2493–2537.

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

[12] Elman, J. L. Finding structure in time. *Cognitive science 14*, 2 (1990), 179–211.

[13] Feifei Li, R. K., and Xu, D. Lecture 10. *CS231n, Stanford University* (2020).

[14] Feng, S., Wallace, E., II, A. G., Iyyer, M., Rodriguez, P., and Boyd-Graber, J. Pathologies of neural models make interpretations difficult. *Empirical Methods in Natural Language Processing* (2018).

[15] Heo, J., Lee, H. B., Kim, S., Lee, J., Kim, K. J., Yang, E., and Hwang, S. J. Uncertainty-aware attention for reliable interpretation and prediction. In *Advances in Neural Information Processing Systems* (2018), pp. 909–918.

[16] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[17] Hoover, B., Strobelt, H., and Gehrmann, S. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276* (2019).

[18] Houck, J., Moyers, T., Miller, W., Glynn, L., and Hallgren, K. Motivational interviewing skill code (misc) version 2.5. *PubMed Central* (2012).

[19] Jurafsky, D., and Martin, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence)*, 1 ed. Prentice Hall, 2000.

[20] Lee, C. M., Neighbors, C., Lewis, M. A., Kaysen, D., Mittmann, A., Geisner, I. M., Atkins, D. C., Zheng, C., Garberson, L. A., Kilmer, J. R., et al. Randomized controlled trial of a spring break intervention to reduce high-risk drinking. *Journal of consulting and clinical psychology 82*, 2 (2014), 189.

[21] Li, J., Luong, M.-T., and Jurafsky, D. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057* (2015).

[22] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002* (2017).

[23] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[24] Loshchilov, I., and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[25] Luong, M.-T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[26] Milad Mohammadi, Rohit Mundra, R. S. L. W. A. K. Lecture notes: V. *CS224n, Stanford University* (2019).

[27] Miller, W., and Rollnick, S. *Motivational Interviewing: Helping People Change*. Applications of Motivational Interviewing Series. Guilford Publications, 2012.

[28] Neighbors, C., Lee, C. M., Atkins, D. C., Lewis, M. A., Kaysen, D., Mittmann, A., Fossos, N., Geisner, I. M., Zheng, C., and Larimer, M. E. A randomized controlled trial of event-specific prevention strategies for reducing problematic drinking associated with 21st birthday celebrations. *Journal of consulting and clinical psychology 80*, 5 (2012), 850.

[29] Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063* (2012).

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

[31] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *openai.com* (2019).

[32] Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822* (2018).

[33] Roy-Byrne, P., Bumgardner, K., Krupski, A., Dunn, C., Ries, R., Donovan, D., West, I. I., Maynard, C., Atkins, D. C., Graves, M. C., et al. Brief intervention for problem drug use in safety-net primary care settings: a randomized clinical trial. *Jama 312*, 5 (2014), 492–501.

[34] Schippers, G., and Schaap , C. The motivational interviewing skill code: Reliability and a critical appraisal. *Behavioural and Cognitive Psychotherapy 33* (07 2005), 285 – 298.

[35] Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[36] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C. V., and Meteer, M. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics 26*, 3 (2000), 339–373.

[37] Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243* (2019).

[38] Sun, C., Qiu, X., Xu, Y., and Huang, X. How to fine-tune bert for text classification? *arXiv preprint arXiv:1905.05583* (2019).

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

[40] Tanana, M., Hallgren, K., Imel, Z., Atkins, D., Smyth, P., and Srikumar, V. Recursive neural networks for coding therapist and patient behavior in motivational interviewing. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality* (2015), pp. 71–79.

[41] Tanana, M., Hallgren, K. A., Imel, Z. E., Atkins, D. C., and Srikumar, V. A comparison of natural language processing methods for automated coding of motivational interviewing. *Journal of substance abuse treatment 65* (2016), 43–50.

[42] Tanana, M. J., Soma, C. S., Srikumar, V., Atkins, D. C., and Imel, Z. E. Development and evaluation of clientbot: A patient-like conversational agent to train basic counseling skills. *Journal of Medical Internet Research* (2019).

[43] Taylor, A., Marcus, M., and Santorini, B. The penn treebank: An overview. *Abeillé A. (eds) Treebanks. Text, Speech and Language Technology, vol 20. Springer, Dordrecht* (2003).

[44] Taylor, W. L. "cloze procedure": A new tool for measuring readability. *Journalism quarterly 30*, 4 (1953), 415–433.

[45] Tollison, S. J., Lee, C. M., Neighbors, C., Neil, T. A., Olson, N. D., and Larimer, M. E. Questions and reflections: the use of motivational interviewing microskills in a peer-led brief alcohol intervention for college students. *Behavior Therapy 39*, 2 (2008), 183–194.

[46] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[47] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[48] Wang, S., and Jiang, J. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905* (2016).

[49] Weizenbaum, J. Eliza - a computer program for the study of natural language communication between man and machine. *Commun. ACM 9*, 1 (1966), 36–45.

[50] Weng, L. Attention? attention! *lilianweng.github.io/lil-log* (2018).

[51] Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE 78*, 10 (1990), 1550–1560.

[52] Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[53] Xiao, B., Can, D., Gibson, J., Imel, Z. E., Atkins, D. C., Georgiou, P. G., and Narayanan, S. S. Behavioral coding of therapist language in addiction counseling using recurrent neural networks. In *Interspeech* (2016), pp. 908–912.

[54] Xiao, B., Imel, Z. E., Georgiou, P. G., Atkins, D. C., and Narayanan, S. S. " rate my therapist": Automated detection of empathy in drug and alcohol counseling via speech and language processing. *PloS one 10*, 12 (2015), e0143055.