IMPROVING LANGUAGE UNDERSTANDING AND SUMMARIZATION BY LEVERAGING AUXILIARY INFORMATION THROUGH SELF-SUPERVISED OR UNSUPERVISED LEARNING.

by

Karan Singla

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

December 2021

Acknowledgments

I would like to thank my advisor Dr. Shrikanth Narayanan for guiding me through this tedious but fun PhD journey and also encouraging me to take on diverse fundamental research problems even when they don't fit our short-term research goals at the time.

I will like to thank my quals and dissertation committee members Dr. Aiichiro Nakano, Dr. Jonathan May and Dr. Morteza Dehghani for being part of my committee and providing valuable feedback. A special thanks to Dr. Dogan Can for being a guiding light and always being my support when I need clarity about research, life and career.

I would like to thank all my colleagues and friends at SAIL. It's hard to name all of you here. I owe all of you a great deal of gratitude for countless discussions and ideas.

I would like to thank my parents, my brother and my sister for being my pillar to help me accomplish this goal. Finally my friends, to name a few: Hannes, Anil, Rens, Tim, Anastasia, Eunyoung, Anubhav, Midhun, Anuj, Vidhi, Silu and many more without whose unwavering support this milestone would have never been possible.

Contents

List of Tables

List of Figures

Abstract

Spoken language understanding (SLU) is an exciting field of research which lies at the intersection between speech and language processing. It investigates human/machine, human/human, or machine/machine communication by leveraging information and technologies from natural language processing, signal processing, machine learning, pattern recognition and artificial intelligence. SLU systems are designed to extract the meaning from a spoken utterance and its applications are vast, from voice search in mobile devices, understanding intentions and behaviors to meeting summarization, attracting interest from both commercial and academic sectors. Understanding these human centered conversational data includes inferring underlying intent and behavior of the speaker.

Existing methods for SLU often require a costly pipeline of systems for understanding a spoken utterance. The typical pipeline based speech processing includes an automatic speech recognizer (ASR), which is used to transcribe speech into text. This text is then used by NLP pipelines for classification or regression tasks. Many different SLU tasks include information understanding, emotion and behavior understanding and use these speech processing pipelines for natural language understanding. However, there have been limited efforts for multimodal understanding of behavior primarily due to unavailability of End-2-End annotations (annotations which are done by listening to speech). Additionally these SLU pipelines fail to efficiently leverage useful auxiliary information from acoustic-prosodic cues, unsupervised clustering and also the gains from joint multitask training with language learning tasks.

In my work, I show that leveraging acoustic-prosodic information can aid lexical text for understanding behavior codes and propose methods for multimodal transcription-free prediction of them. I also propose novel methods for leveraging auxiliary information for learning text representations and summarization. First, I show that learning generic task representations which exploit additional monolingual data using joint multitask training can help generalize task related knowledge to a bigger vocabulary. Second, I propose factored extractive summarization techniques which can efficiently summarize spoken language by exploiting psycholinguistic information and topic models. We believe this provides enough evidence that End-2-End methods which leverage linguistic structure and exploit auxiliary information using self-supervision techniques enable multi-modal transcription-free understanding of behavior and efficiently summarize spoken language.

Chapter 1

Introduction

Spoken language understanding (SLU) is an exciting field of research which lies at the intersection between speech and language processing. It investigates human/machine, human/human, or machine/machine communication by leveraging information and technologies from natural language processing, psychology, signal processing, machine learning, pattern recognition and artificial intelligence. SLU systems are designed to extract the meaning from a spoken utterance and its applications are vast, from voice search in mobile devices, understanding intentions and behaviors to meeting summarization, attracting interest from both commercial and academic sectors (Tur & De Mori, 2011). Understanding these human centered conversational data includes inferring underlying intent and behavior of the speaker.

When it comes to making automatic understanding systems. We typically rely heavily on costly data resources in form of annotated corpus for classifcation and regression (De Mori et al., 2008). Spoken language understanding systems generally require big speech and text processing pipelines. These pipelines also rely on costly tools, including but not limited to Machine Translation (MT) (Newmark, 1998; Servan et al., 2010; Jabaian et al., 2010), Automatic Speech recognition (ASR) (De Mori et al., 2008; Tur & De Mori, 2011; Cheung et al., 2017). These modules generally rely on expensive parallel corpus (bilingual sentence pairs) and transcribed corpus for speech segments respectively (Koehn, 2009; Lee, 1988; Povey et al., 2011). There is a ever-growing interest in removing dependency on costly resources and leveraging auxiliary information cheaply available in-form of additional

Natural Language processing

Learning text representations: *parallel data*, *monolingual data*, *domain data*, *unsupervised (mBERT)*
Text classification: *annotated data*, *clustering*, *knowledge bases*, *unsupervised*
Entity recognition: *annotated data*, *knowledge bases*, *unsupervised*
Summarization: *summary-document pairs*, *knowledge bases*, *extractive*, *unsupervised*

Spoken language understanding

Automatic speech recognition: *transcribed data*, *read data*
Speaker embeddings: *speaker labels*, *unsupervised*
Utterance classification: *transcribed data*, *acoustic-prosodic cues*, *speech annotated data*

Figure 1.1: Some applications and data needs based on literature review. Red mention costly resources/methods, green mentions cheap resources and yellow mentions methods wich are doubtful

or pretraining data resources like monolingual text, knowledge graphs and read speech (Mikolov et al., 2013; Peters et al., 2018; Pan et al., 2017; Schneider et al., 2019).

Auxiliary as defined in the dictionary is something which is secondary or supplementary, or something with subordinates or assists[1]. There are many ways we use auxiliary information to understand and build systems for automatic spoken language understanding a task. Popularly used techniques like self-supervised or unsupervised learning be it for text or be it for speech representations, generally exploit inherent linguistic structure of the spoken language (Kidd et al., 2015). This inherent structure, is what I believe is a semantic breakdown of how we interpret language, where we learn lower level representations or summarization of words-in-context to understand sentences (Peters et al., 2018), interpret sentences to make sense of a document (Yang et al., 2016) and so on.

In practice, existing methods for SLU often require a costly pipeline of systems for understanding a spoken utterance. The typical pipeline based speech processing

[1] https://languages.oup.com/google-dictionary-en/

includes an automatic speech recognizer (ASR), which is used to transcribe speech into text. This text is then used by NLP pipelines for classification or regression tasks. Many different SLU tasks, including but not limited to speaker state understanding (in form of behavior or emotion) or understanding the lexical meaning (meaning of language units: phonemes, words, sentences) use these speech processing pipelines for spoken language understanding. More recently modeling techniques which use self-supervised or unsupervised modeling which exploit inherent linguistic structure reduce the problem of data sparsity which mainly happened due to limited annotated data (Haghani et al., 2018; Serdyuk et al., 2018). Moreover, these methods which require minimal supervision can also potentially handle ambiguities introduced due to pipeline based approach to SLU .

There has been active research in multimodal (using text and speech) understanding of intent (which requires understanding of meaning) (Tur & De Mori, 2011; Desot et al., 2019; Sharma et al., 2021) and emotion which needs understanding of speaker state (Sebe et al., 2005; Schuller et al., 2002; Siriwardhana et al., 2020). Marechal et al. (2019) provides a good overview of SLU for multimodal emotion understanding. Recent modeling techniques which use self-supervised or unsupervised modeling require minimum human supervision reduce the problem of data sparsity which happens due to limited annotated data (Tafforeau et al., 2016; Qian et al., 2017; Baevski et al., 2020). Moreover, these methods which require minimal supervision can also potentially handle ambiguities introduced due to pipeline based approach to SLU . However there have been limited efforts for multimodal understanding of behavior which is generally credited to account for both speaker-state and meaning . This is primarily due to unavailability of readily available behavior coded data and lack of End-2-End annotations (annotations which are done by listening to speech). Thus failing to capture auxiliary information e.g: acoustic-prosodic patterns which can help with better and more transparent understanding of behavior from speech.

Behavior is systematically studied in psychotherapy process research (Hardy & Llewelyn, 2015). MISC behavior coding manual originally proposed in the context of Motivational interviewing has been over the years extended to multiple domains of psychotherapy like Cognitive Behavioral therapy and more recently client language rating (Miller et al., 2003; Glynn & Moyers, 2012). Behavior codes which are similar to dialog acts in NLP literature have been widely used and adopted to understand the process of psychotherapy, evaluate gains and also train & evaluate psychotherapists. Behavior code annotations are done by listening to the utterance segmented speech utterances. Therefore taking into account both acoustic-prosodic, speaker-related and lexical-text to annotate a behavior code for each spoken utterance in a didactic conversation between a therapist and a patient.

Figure 1.2 gives a high level overview of my contributions and SLU or Spoken language classification (SLC) modeling techniques. In the first part of my work, I focus on automatic understanding of multi-modal transcription-free prediction of behavior codes which require understanding of both lexical and acoustic-prosodic semantics of a spoken utterance. Results indicate that variation in acoustic-prosodic variation between words and sentences can help with better predictions and provides more clarity on the effect of acoustic-prosodic in understanding behavior codes (Singla et al., 2018; Chen et al., 2019). I propose a novel method for transcription-free behavior coding which follows the premises that breaking a spoken utterance into meaningful units (words) can remove the dependency on making costly tools like ASR (Singla et al., 2020). I believe moves us towards a nearing future where we can just listen to a spoken utterance, annotate and predict. There are multiple

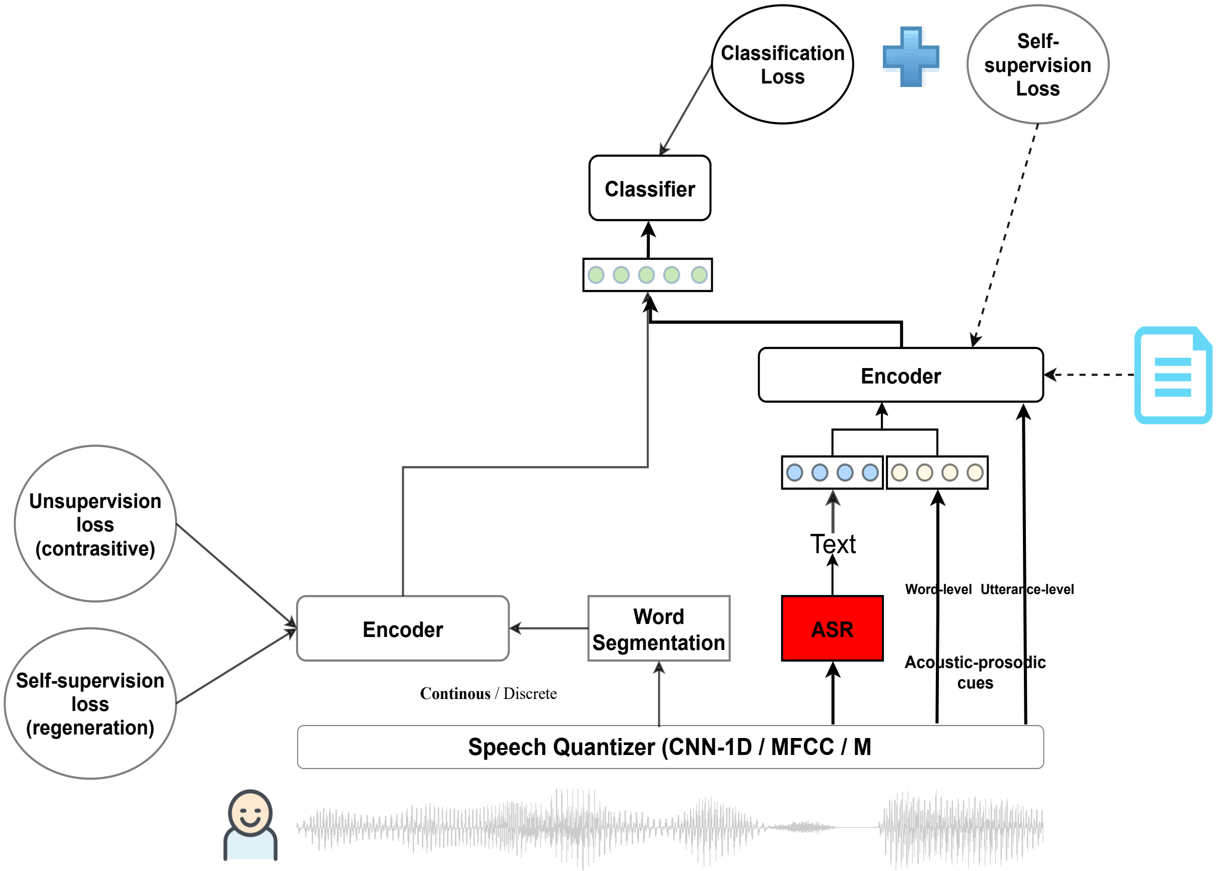


Figure 1.2: Some applications and data needs based on literature review. Red mention costly resources/methods, green mentions cheap resources and yellow mentions methods which are doubtful

benefits to this approach: 1) Better capture ambiguities, 2) Less infrastructure 3) less redundant annotations.

However most SLU system pipelines as it exists still favour using an ASR primarily due to rwo reasons: 1. Ease of work to avoid making many end-2-end systems 2. End-2-End annotations are not available for the majority of tasks. These systems which use lexical text for SLU are still heavily reliant upon human annotations for extracting meaning and understand speaker state. These human annotations are hard to obtain and generalize over a larger information space. Therefore, in the second part of my work I propose methods for leveraging auxiliary information for learning text representations and summarization. Firstly, I show joint multitasking with an auxiliary task of learning contextualized word representations can help with learning high quality multilingual text representations and document classification. Secondly I show use of psycholinguistics norms and unsupervised topic modeling for extractive text summarization. Results show evidence that our system can in-fact transfer information between two tasks: 1. Task specific sentence/document representations and 2. Auxiliary task of learning language by either predicting context or by regenerating the input. Results show evidence (1.5) this joint multitask information can in-fact help task specific layers to generalize over a bigger vocabulary.

Rest of the chapter is divided as follows: first, I describe my contributions on automatic multi-modal transcription free behavior coding, second I describe my work on leveraging monolingual text for learning text representations and factored automatic summarization. I then conclude this chapter by briefly discussing the scope of my thesis.

1.1 Automatic Behavior Coding

In psychotherapy research, behavioral coding is used for identifying and codifying the behaviors which are most relevant to the aims of therapy (Heyman et al., 2014). The objective of this procedure is to define clear and broadly applicable behavioral 'codes' which represent target behavioral constructs that are of interest to a particular study or line of inquiry. Behavioral observation and coding is common practice in many subfields of psychology including diagnosing autism (Pruette, 2013), family and marital observational studies (Margolin et al., 1998; Christensen et al., 2004), and several forms of psychotherapy (Miller & Rose, 2009;

Creed et al., 2016). Manual behavioral coding is costly and time-intensive and there many methods have proposed methods for automatic prediction of them. Most existing methods use lexical text human-transcribed or transcribed using an ASR for predicting behavior codes (Xiao et al., 2016d; Tanana et al., 2016; Pérez-Rosas et al., 2017; Gibson et al., 2017b, 2019) Thus, depending upon an expensive tool like ASR for transcription and moreover also loosing out on acoustic-prosodic cues which provide discriminatory information for automatic understanding and prediction of behavior codes.

1.1.1 Exploiting Acoustic-prosodic Cues

I present an approach for predicting utterance level behaviors in psychotherapy sessions using both speech and lexical features. We train long short term memory (LSTM) networks with an attention mechanism using words, both manually and automatically transcribed, and prosodic features, at the word level, to predict the annotated behaviors (Singla et al., 2018; Chen et al., 2019). I demonstrate that prosodic features provide discriminative information relevant to the behavior task and show that they improve prediction when fused with automatically derived lexical features. Additionally, I investigate the weights of the attention mechanism to determine words and prosodic patterns which are of importance to the behavior prediction task. Chapter 2 shares more insights and details about this contribution.

1.1.2 Transcription-free Behavior Coding

Spoken language understanding tasks usually rely on pipelines involving complex processing blocks such as voice activity detection, speaker diarization and Automatic speech recognition (ASR). I propose a novel framework for predicting utterance level labels directly from speech features, thus removing the dependency on first generating transcripts and doing transcription free behavioral coding. Our classifier uses a pretrained Speech-2-Vector encoder as bottleneck to generate word-level representations from speech features. This pretrained encoder learns to encode speech features for a word using an objective similar to Word2Vec. My proposed approach just uses speech features and word segmentation information for predicting spoken utterance-level target labels (Singla et al., 2020). I show that our model achieves competitive results to other state-of-the-art approaches which use transcribed text for the task of predicting psychotherapy-relevant behavior codes.

1.2 Text Summarization

Second part of my work focuses on understanding meaning of spoken language. In this I only focus on leveraging auxiliary information for understanding semantics by exploiting inherent linguistic structure. We propose a joint multitask approach for leveraging monolingual text for learning task specific text representations. Second, which is an ongoing work, we propose factored extensions of some traditional extractive summarization algorithms like: Luhn's, MMR, Text Rank.

1.2.1 Text Representation

It's a general convention to pretrain a deep neural network for a generic auxiliary task before it's fine-tuned using the task related related data. This pre-training is generally done to get a good initilization point as the the mdeol learns to encode essential information using this auxiliary task used for pre-training. This pre-training can be different for different applications, for e.g. some form of Language modeling for text or speech understanding, input or context regeneration, etc. Pretrained model is then used a building block to perform a high level task.

One widely used trend is to fine-tune the entire model or few layers of the pretrained model to the task related data. We believe that pretraining followed by fine-tuning is beneficial in most cases but our experiments show it also possibly leads to overfitting. Where fine-tuning can lead to loss of generality and hence, loss of the benefits of the pre-trained model. It's possible that the model is only fine-tuned to work on small information space it is fine-tuned on.

We will explain this process using a bubble diagram of information spaces. We define 3 information spaces:

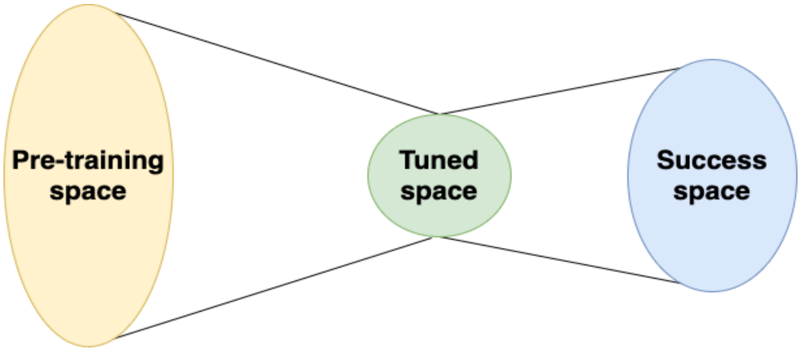


Figure 1.3: Information spaces

- **Pre-training space:** The information input space on which model is pre-trained on. It's generally very large.
- **Success space:** It's the space which only has useful information from pre-training space needed for the task. A subset of pretraining space.
- **Tuned space:** It's a subset of both pre-training and success space. Ideally, we want tuned space to be similar to success space.

In Figure 1.3 we hypothesize that fine-tuning can lead to model being tuned to a small subspace (*Tuned space*), which is smaller than the actual intended *Success space* needed for the model to generalize and understand the task. In this thesis, we propose methods to stretch this *Tuned space* and see, how it can look more closer to *Success space*. We hypothesize that using these methods to leverage auxiliary information lead to a stretched Tuned space shown in Figure 1.4

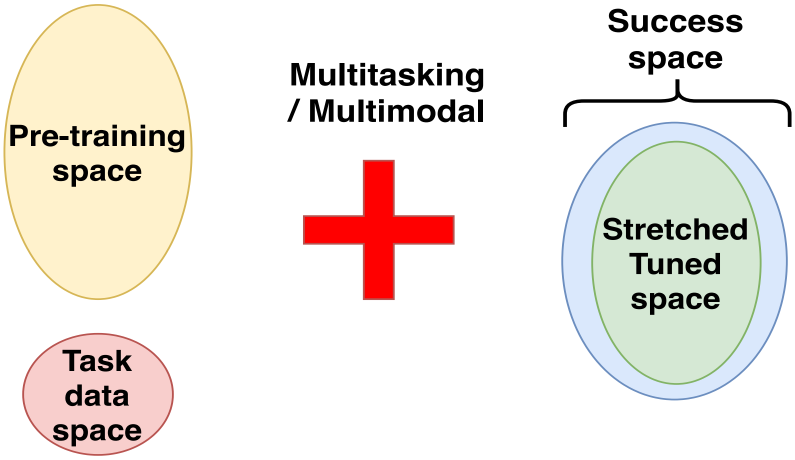


Figure 1.4: Stretched tuned space

Based on described underlying motivation I present a novel multi-task modeling approach to learn multilingual distributed representations of text. Our system learns word and sentence embeddings jointly by training a multilingual skip-gram model together with a cross-lingual sentence similarity model (Singla et al., 2018). This architecture can transparently use both monolingual and sentence aligned bilingual corpora to learn multi-lingual embeddings, thus covering a vocabulary significantly larger than the vocabulary of the bilingual corpora alone. Proposed model shows competitive performance in a standard cross-lingual document classification task. I also show the effectiveness of our method in a limited resource scenario.

Figure 1.5 show the benefit of learning a multilingual sentence encoder and word representation task in a multilingual fashion. This word embedding task can use additional monolingual data. Hence transferring information between two tasks and helping multilingual information to transfer over a bigger vocabulary. Not just text representation, but also spoken language understanding (like: spoken utterance classification) suffers from the dependency of using annotated data. This annotated data can be limited or can lack essential information that can be beneficial for prediction, for e.g: predicting a label for a spoken utterance based on transcriptions. Hence, missing information that can be provided by acoustic-prosodic patterns.

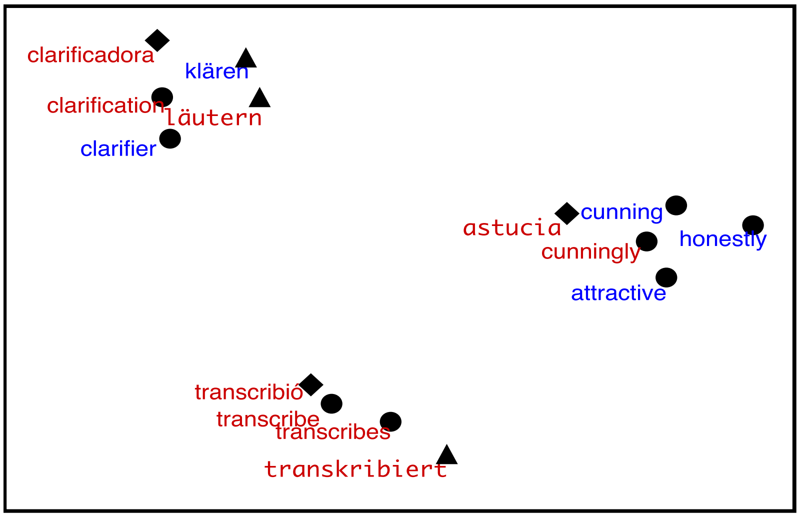


Figure 1.5: t-SNE projections for 3 English words (clarification, transcribe, cunningly) which are not in the parallel corpus and their four nearest neighbors. Red words are only in the monolingual corpus. Blue words exist in parallel corpus too.

Additionally, I propose a novel approach of multitask learning for an end-to-end optimization technique for document classification. The application motivation comes from the need to extract "Situation Frames (SF)" from a document within the context of DARPA's LORELEI program targeting humanitarian assistance and disaster relief. We show the benefit of our approach for extracting SF: which includes extracting document types and then linking them to entities. We jointly train a hierarchical document classifier and an auto-encoder using a shared word-level bottleneck layers. Our architecture can exploit additional monolingual corpora in addition to labelled data for classification, thus helping it to generalize over a bigger vocabulary. We evaluate these approaches over standard datasets for this task. Our methods show improvements for both document type prediction and entity linking (Singla & Narayanan, 2020).

1.2.2 Factored Extractive Summarization

Extractive summarization methods extract parts of text verbatim that are deemed informative to create a summary. However, existing extractive methods rely only lexical words for addressing the problem of redundancy and diversity of information. in this chapter, we propose extensions of widely used extractive alogorithms, namely: Luhn, MMR (Max marginal relevance) and Textrank whichcan incorporate information from not just lexical context but also other factors. Our results indicate that use of factors like psycho-linguistics and unsupervised topic models can help to improve quality and provide better context for the summaries. Our results on a standard meeting corpus suggest that these factors help generate more relevant summaries. Our proposed extension while allows a user to control contribution of each information type improves over some of the well-known methods for both automatic and manual evaluation metrics.

1.3 Scope of the Thesis

As in today's world, a large part of AI applications that involves understanding text also includes understanding spoken language. We show that acoustic-prosodic information e.g.: pitch variations, pauses, etc. can complement information already provided from lexical text. We show results for proposed methods in the context of predicting behaviour codes, however, we believe that these methods can be applied to other SLU applications.

Our results also indicate that text and acoustic-prosodic information have some overlap and not necessarily complementary. Therefore, we propose a method for learning an end-2-end system for predicting behavior codes directly from speech features. This shows that high quality behavior code prediction can be done with any transcriptions but only word boundaries. We believe that it is important not just for machines but also for humans to segment speech into words for understanding. Our current model uses word boundaries provided by a force-aligner, however, we believe that it is possible to learn high quality word segmentation information in an unsupervised manner. This is an ongoing work and we plan to address this in the future work. We also plan to evaluate our proposed methods on additional tasks: for e.g. emotional state prediction, sentiment prediction and topic modeling.

Additionally we show that for learning text representations auxiliary information is not just helpful for pretraining but can also help the system to generalize over a bigger input space. However in order to see gains it's important that the auxiliary task is complementary to the intended task and not in competition. For e.g. When we try to do document classification we want sentence auto-encoder to help document classification but not stop it from learning. We believe proposed methods can be promising directions to learn text representations systems using limited amount of costly supervised corpus.

Chapter 2

Leveraging acoustic-prosodic information

In this chapter, we present an approach for predicting utterance level behaviors in psychotherapy sessions using both speech and lexical features. We train long short term memory (LSTM) networks with an attention mechanism using words, both manually and automatically transcribed, and prosodic features, at the word level, to predict the annotated behaviors. We demonstrate that prosodic features provide discriminative information relevant to the behavior task and show that they improve prediction when fused with automatically derived lexical features. Additionally, we investigate the weights of the attention mechanism to determine words and prosodic patterns which are of importance to the behavior prediction task.

Index Terms: prosody, mutlimodal learning, behavioral signal processing

2.1 Introduction

Both the words that are spoken and the way in which they are spoken are of fundamental importance in psychotherapy conversations. There are many studies demonstrating the importance of the lexical channel for predicting behaviors in psychotherapy Can et al. (2012, 2015); Xiao et al. (2016b), but multimodal information like visual and acoustic cues also carry a wealth of information that is potentially complimentary to the lexical modality Narayanan & Georgiou (2013), and has received less attention in this domain.

We focus on data from Motivational Interviewing (MI) sessions, a type of psychotherapy focused on behavior change. Behavior is generally monitored and codified in the form of behavioral coding, which is the process of a human manually observing a session and annotating the behaviors of the participants in that session, as defined by a coding manual. The Motivational Interviewing Skill Code (MISC) manual defines both session and utterance level behaviors that are of interest for understanding therapist efficacy in MI Miller et al. (2003). Several approaches have been proposed for automating the behavioral coding procedure to predict gestalt session level behaviors, especially therapist empathy, using lexical information (both manually and automatically derived) Xiao et al. (2015b), speech rate entrainment Xiao et al. (2015a), and prosody Xiao et al. (2014). At the utterance level, automating the behavioral coding process has been entirely focused on linguistic features Xiao et al. (2016b); Tanana et al. (2016); Pérez-Rosas et al. (2017). In this work, we inspect: if utterance-level behavior codes can be predicted using prosodic cues; and if prosodic information can assist lexical information in making better predictions of participants' behaviors.

We hypothesize prosodic information such as variation in pitch, loudness, pause, etc. will have an important role in predicting behaviors in psychotherapy and can assist lexical features for making improved predictions. Therefore, we propose a multimodal approach for predicting behavior codes that exploits both prosodic and lexical information at the word level. We show that prosodic information can assist lexical information in making a multimodal prediction. Our multimodal architecture is largely inspired from Gibson et al. (2017a) and Xiao et al. (2016b). Thus, we use Bidirectional long short term memory (LSTM) networks with a self-attention mechanism for predicting MISC Codes at the level of utterances using multimodal information i.e prosodic and lexical features. Our network is different from prior research, as we use a unified architecture, i.e., the same model for predicting therapist/client codes.

2.2 Related Work

Several computational models have been proposed for predicting MISC behavioral codes at the utterance level Can et al. (2015); Xiao et al. (2016b); Tanana et al. (2016). Researchers have addressed this problem by using variety of features, such as word n-grams and linguistic features Can et al. (2012) and recurrent neural networks (RNNs) with word embedding features Xiao et al. (2016b); Gibson et al. (2017a). Methods using RNNs have shown superior performance to other models (e.g., MaxEnt) for utterance level behavioral code prediction Xiao et al. (2016b). The success of these RNN based models demonstrates that learning in-domain word representations and parameters in an end-to-end fashion offers better modeling for this task. These models typically use separate models for therapist and client codes, whereas we propose a unified model which still uses utterance level speaker information.

Self-attention mechanisms, which enable models to attend to particular words based on input for predicting output classes, have been used widely in natual language processing Bahdanau et al. (2014); Yang et al. (2016); Vaswani et al. (2017) and speech processing Chorowski et al. (2015). Recently Gibson et al. (2017a) extended the work from Xiao et al. (2016b) by using a self-attention mechanism for predicting utterance level MISC codes. They show how attention can improve the interpretability and help in better understanding the decisions made by the model.

While using multimodal information is rather unexplored in predicting utterance level MISC codes, it has been an exciting venue of research in some other related domains such as multimodal parsing Tran et al. (2017), prediction of psychological disorders Yu et al. (2013), and audio-visual applications like speech recognition Mroueh et al. (2015). Our proposed multimodal approach is similar to Tran et al. (2017) in the sense that we also concatenate prosodic features obtained from audio signals with lexical features to get word-level representations.

2.3 Data

Code	**Description**	**#Train**	**#Test**
Therapist			
REF	Reflection	6577	3456
QES	Question	6546	3348
OTH	Other	13112	7625
Total		26235	14429
Client			
FN	Follow/Neutral	22020	12229
NEG	Sustain Talk	4019	1660
POS	Change Talk	3151	1272
Total		29190	15161

Table 2.1: Frequency of samples for each class label in the preprocessed MISC data.

In this chapter, we use data from Motivational Intervewing sessions, presented in Atkins et al. (2014); Baer et al. (2009), for behavior prediction. Table 2.1 shows statistics of utterance level MISC data used in this chapter after removing utterances where there is a speaker overlap. For Therapist codes, a reflection (REF) is a reflective listening statement made by the counselor in response to a client statement. Question (QES) is either an open or a close question asked by the therapist. Other (OTH) can include Advise, Affirm, Confront, Facilitate etc. among other therapist behaviors. Client behavior is observed from three dimensions. In a follow-neutral turn (FN), there is no indication of client inclination either toward or away from the target behavior change. Client behavior is otherwise marked with a positive (POS) or negative (NEG) valence, depending on whether it reflects inclination toward (POS) or away from (NEG) the target behavior change. Figure below shows a snapshot for our data pipeline.

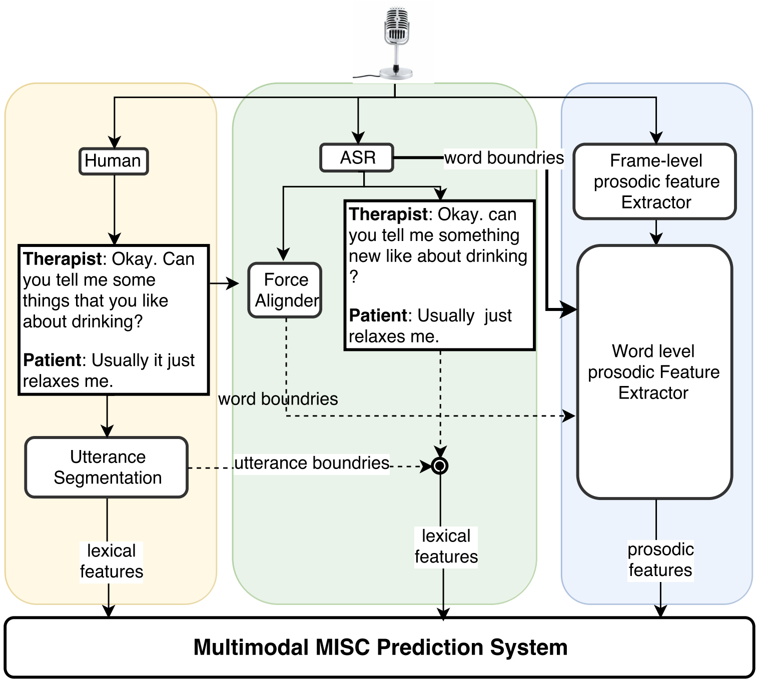

Human
ASR
word boundries
Frame-level prosodic feature Extractor
Therapist: Okay. Can you tell me some things that you like about drinking?
Patient: Usually it just relaxes me.
Force Aligndner
Therapist: Okay. can you tell me something new like about drinking ?
Patient: Usually just relaxes me.
Word level prosodic Feature Extractor
word boundries
Utterance Segmentation
utterance boundries
lexical features
lexical features
prosodic features
Multimodal MISC Prediction System

Figure 2.1: Data Pipeline. Left part shows data flow to get human transcribed utterance level data. Middle part shows the data flow to create similar data using automatic transcription. Right, shows pipeline for extracting prosodic features at the word level.

Our data flow pipeline handles three main types of data:

- **Human transcribed text data:** Audio signals are first transcribed at speaker turn level by humans. Each turn is then segmented by humans experts to get utterances. Human experts then annotate these Therapist and Patient utterances with MISC codes.

- **Automatically transcribed text data:** Middle part of Figure 1 shows the pipeline where we use utterance text generated using ASR. We use the ASR system presented in Xiao et al. (2016c) to get automatic transcripts from audio signals. ASR does use speaker turn boundaries marked by human transcribers. Reported word error rate (WER) is 44.1 % Xiao et al. (2016c), where a major chunk of errors is because of substitution (27.9 %). We use utterances segmentation information and MISC labeling done by human experts for generating this data.

- **Word level Prosodic feature Extractor :** Using audio signals we first extract prosodic features at frame level. As our multimodal approach uses word-level features, for human transcribed training data we use a force aligner Xiao et al. (2016c) to align human transcribed transcripts to get word boundaries. For automatic generated text data, ASR directly gives word boundaries to extract word-level prosodic features. Prosodic features extraction is described in more detail in the next section.

2.4 Method

2.4.1 Features

Our multimodal system can exploit two types of features; namely features exploiting prosodic information and lexical text.

Prosodic

- **Prosodic (a):** We use pitch, loudness and jitter as prosodic features. We extract frame-level pitch using pyaudioanalysis Giannakopoulos (2015) &

loudness and jitter using *Praat* Boersma (2006), where frame size is 50ms and step size is 25ms. We then calculate the mean and standard deviation (std) across frames within a word to represent 6 (3 mean, 3 std) word level features.

- **Pause (p):** We also encode word-level pause information i.e pause taken before and after a word. For each word, pause is quantized into a 10 bit vector (5 for pause before and 5 for pause after) depending upon if pause time lies before, between and after $\{0.01, 0.06, 0.2, 1.0\}$ seconds. These boundaries are selected so that the words are approximately uniformly distributed in those bins.

- **Average word length (wl):** We keep an additional feature for marking word length i.e number of frames used to speak a word. We normalize word length by average word length over the entire train and test dataset separately.

We concatenate a_i, p_i and wl_i to get a 17-dimensional prosodic feature representation A_i for each word W_i.

Speaker normalization: There are various different studies collected across different settings that are part of this dataset with different speakers, both in terms of therapists and patients. Therefore, we do a two-fold speaker normalization. First, we do a z-normalization for each audio feature for each study type and second, we normalize each audio feature for each speaker (Therapist and Patient) for each audio session.

Lexical

For textual features we remove all punctuations and lower case all words. We also replace any words having frequency less than 5 with the <unk> symbol.

Our final vocabulary has 11219 unique words. Each word W_i is represented by a 100-dimensional vector T_i, initialized using a uniformly distributed random word embedding layer which we learn as a part of the encoder described in the following section.

2.4.2 Utterance Encoder

We assume that each utterance is represented by a word sequence $W = \{W_0, W_1, \cdots, W_{L-1}\}$, where L is the number of words in the utterance. Each word can be represented either by prosodic features, or by lexical text, or both. We then assume there exists a function $c = f(W)$ that maps W to a behavioral code $c \in 1, 2, \cdots, C$, with C being the number of defined code types. Our goal is to find the function $f*$ minimizing the error between the predicted and expert-annotated codes.

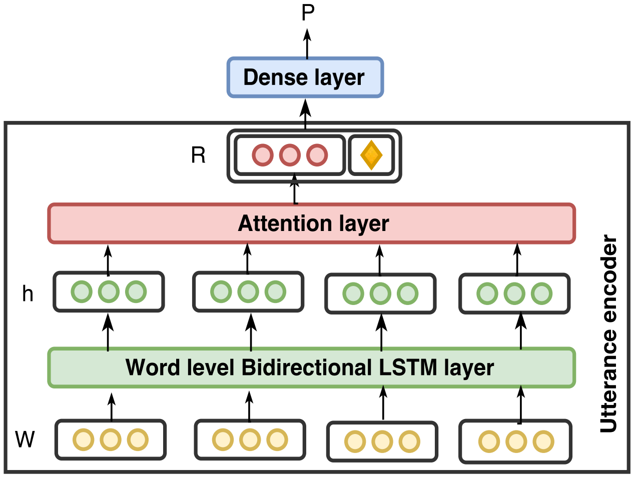


Figure 2.2: Architecture for Utterance Encoder. ♦ can be 1 or 0 for Therapist and Patient utterance respectively.

We use a parametric composition model to construct utterance-level embeddings from word-level embeddings. We process word-level embeddings with an

LSTM Hochreiter et al. (2001); Hochreiter & Schmidhuber (1997) and then take a weighted average of the LSTM outputs using a task-specific attention model Yang et al. (2016). There are various implementations of LSTMs available; in this work we use an implementation based on Zaremba et al. (2014). The LSTM outputs (hidden states) h_i contextualize input word embeddings W_i by encoding the history of each word into its representation. The attention layer can be seen as a mechanism for accessing internal memory of the system, i.e. the hidden states of the LSTM. It can learn what to retrieve from the memory while constructing an utterance representation. For example, it can learn to ignore stop-words or downweight words that are not essential for predicting behavioral codes. We use an attention layer (equations 1-3) with an internal context vector Yang et al. (2016).

$$k_i = \tanh(W h_i + b) \tag{2.1}$$

$$\alpha_i = \text{softmax}(k_i^T a) \tag{2.2}$$

$$R = \sum_i \alpha_i h_i \tag{2.3}$$

The attention layer first applies a one-layer MLP to its inputs h_i to derive the keys k_i. Then it computes the attention weights by applying a softmax nonlinearity to the inner products between the keys k_i and the internal context vector a . Finally it computes the sentence representation R by taking a weighted average of its inputs. The context vector a is a model parameter that is initialized with uniform weights so that it behaves like an averaging operation at the beginning of the training.

We then concatenate oracle speaker information (Therapist (1) vs Client (0)) to R before it's passed through a dense layer to get P. P is a C-dimensional vector on which we take softmax to predict MISC label. We also show experiments with the model where the utterance encoder doesn't use the attention mechanism. Instead, it just uses the last hidden state from the LSTM layer (*LSTM-l*). We will refer to our model with attention as *LSTM-a*.

2.4.3 Multimodal Approach

Prosodic feature vector A_i for each word W_i is first processed through a dense layer to get a high dimensional representation, which matches the lexical representation of the word in terms of dimensionality before it's fed into the LSTM layer.

We do a multimodal combination by two methods :

- **Comb-WL** : Word-level lexical features T and prosodic features A are word-wise concatenated to make input W before feeding it to the utterance encoder for predicting MISC labels.

- **Comb-LF** : As show in Figure 2.3, we first train utterance encoder using lexical features and a separate encoder using prosodic features. For fusion, word-level audio sequence W_A is processed through a pretrained utterance encoder trained on audio data and similarly W_T is processed separately to get R_A and R_T respectively. R_A and R_T are then concatenated and then passed through another dense layer to get the C-dimensional output P. This allows us to tune the entire system for multimodal information in an end-to-end fashion

Training Routine : The batch size is 40 utterances. LSTM hidden state dimension is 100 (50 forward, 50 backward). We use dropout at the embedding layer with drop probability 0.3. Dense layer is of 100 dimensions. The model is

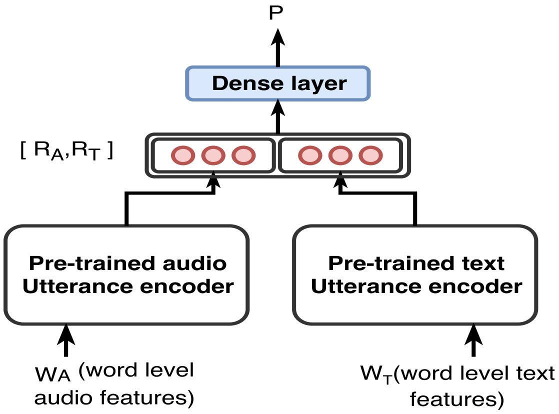

P
Dense layer
[R_A,R_T]
Pre-trained audio Utterance encoder
Pre-trained text Utterance encoder
W_A (word level audio features)
W_T(word level text features)

Figure 2.3: Multimodal system architecture using Comb-LF approach.

trained using the Adam optimizer Kingma & Ba (2014) with a learning rate of 0.01 and an exponential decay of 0.98 after 10K steps (1 step = 40 utterances). We weight each sample using class weights derived using class frequencies. Formally, the weight given to a sample belonging to class i is

$$w_i = \frac{\tilde{w_i}}{\sum_i \tilde{w_i}}, \text{ where } \tilde{w_i} = \frac{\text{total \#samples}}{\text{\#samples}_i}$$

2.5 Experiments & Results

2.5.1 Behavior (MISC) Code Prediction

Baselines

We train models with just lexical features (*Text*) and just prosodic information (*Prosodic*). The first two rows in Table 2 show class averaged f-scores for our baseline systems where we only use one modality (lexical-features or prosodic-features). Model with just lexical features (*Text*) performs better than the model which only uses word-level prosodic information (*Prosodic*). *Prosodic* model performs better than majority class baseline, which is 0.33, since we report class averaged f-scores.

This shows that prosodic information alone is quite informative about predicting behavior codes.

Human vs Automatically transcribed Data

Bottom part of Table 2.2 shows that multimodal information can in fact help in making a better prediction of behavior codes compared to single modality models (*Text* and *Prosodic*). We get best results for *Comb-LF* where we do late fusion of utterances. Scores where we use attention are better, therefore, we use model with attention (*LSTM-a*) for further experiments.

Features	**Avg. F1-score**	
	LSTM-l	LSTM-a
Text	0.54	0.57
Prosodic	0.42	0.42
Comb-WL	0.56	0.58
Comb-LF	0.58	0.60

Table 2.2: Results for single modality (Text, Prosodic) and multimodal approach for human generated test data.

Using automatically generated lexical features, results in Table 2.3 show high gains for multimodal systems. Comb-LF outperforms other models with automatically transcribed lexical features. This number is a bit worse than the model which uses human transcribed lexical features *Text*. Moreover, the Comb-WL model which fuses word level lexical and prosodic information also shows improvements.

Features	**Avg. F1-score**
ASR text	0.47
Comb-WL	0.52
Comb-LF	0.53

Table 2.3: Results for using automatically generated transcripts from ASR

2.5.2 Attention Weight Analysis

Prosodic model generally gives high weight to utterance endings, indicating it's important to attend to the last part of the utterance for predicting behavior. It reinforces the hypothesis that pitch rises at the end of questions which makes it an important marker for discrimination. It also always gives some weight to start of the utterance, along with attending a bit to word *Did* for example in Figure 4. It can also be seen that *Text* model attends to lexical words that are necessary to mark a question. (high weights to words: did, you, say).

2.5.3 Evaluating on Utterances > 15 words

Ablation experiments where we only choose utterances longer than 15 words (4824 and 5313 samples for Therapist and Patient codes respectively), suggest that *Prosodic* model shows improved performance for longer utterances. Table 2.4 shows results for this. Scores for *Prosodic* features improve only evaluated for longer utterances. ASR text follows a similar trend. Results for combination experiments are also slightly better when evaluated for longer utterances.

These results validate our hypothesis that as prosodic features (pitch, loudness and jitter) are continuous values, what we essentially measure is the variation in them as we pass over words. When the utterance has very few time stamps (less words), the model with prosodic information performs badly as it is not able to cover the variation in them.

2.6 Conclusions and Future Work

in this chapter, we demonstrated that using prosodic features in addition to lexical features aid in the prediction of certain utterance-level behaviors in

Features	**Avg. F1-score**
Prosodic	0.48
ASR text	0.50
Comb-WL	0.54
Comb-LF	0.55

Table 2.4: Ablation experiments results when evaluated on utterances longer than 15 words.

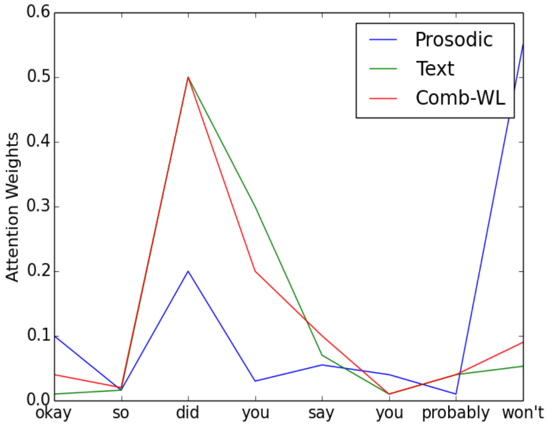


Figure 2.4: Comparison of attention weights for one question sample (QES)

psychotherapy sessions. We employed bi-directional LSTMs with an attention mechanism with both word-level and utterance-level fusion of prosodic and lexical modalities. We also presented an analysis with examples of the types of words and prosodic patterns that are attended to by the attention mechanism. Additionally, we discussed how the length of utterances influences performance of the prosodic modality. Ablation experiments suggest our encoder architecture relies on variation between prosodic features over words; thus, we plan to investigate using discrete representation of prosodic features in the future. We also plan to use more complicated compositional models to represent word-level prosodic information instead of using just the mean and standard deviation.

2.7 Acknowledgements

This work was supported by the NIH.

Chapter 3

Transcription-free Behavior Coding

Spoken language understanding tasks usually rely on pipelines involving complex processing blocks such as voice activity detection, speaker diarization and Automatic speech recognition (ASR). We propose a novel framework for predicting utterance level labels directly from speech features, thus removing the dependency on first generating transcripts and doing transcription free behavioral coding. Our classifier uses a pretrained Speech-2-Vector encoder as bottleneck to generate word-level representations from speech features. This pretrained encoder learns to encode speech features for a word using an objective similar to Word2Vec. Our proposed approach just uses speech features and word segmentation information for predicting spoken utterance-level target labels. We show that our model achieves competitive results to other state-of-the-art approaches which use transcribed text for the task of predicting psychotherapy-relevant behavior codes.

3.1 Introduction

Speech interfaces have seen a widely growing trend and this has brought about increasing interest in advancing computational approaches to spoken language understanding (SLU). Tur & De Mori (2011); Xu & Sarikaya (2014); Yao et al. (2013); Ravuri & Stolcke (2015). SLU systems often rely on Automatic speech

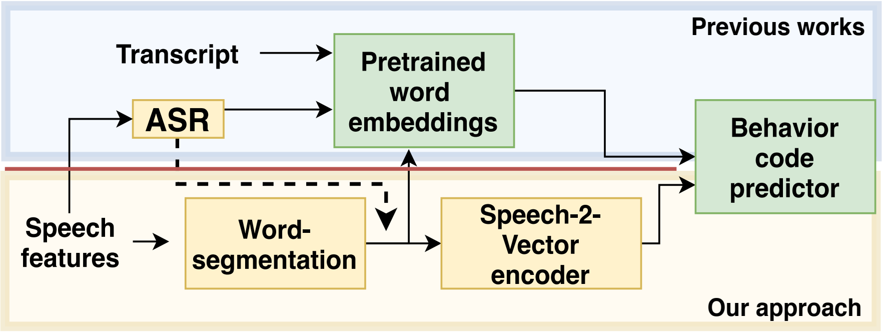


Figure 3.1: Upper part describes all existing approaches which either use ASR or manual transcripts. Lower part shows our proposed approach where we predict behavior codes without using transcripts

recognition (ASR) for generating lexical features. The ASR output is then used for the target natural language understanding task. Furthermore, end-2-end SLU systems for various applications, including speech synthesis Oord et al. (2016), ASR tasks Amodei et al. (2016); Chan et al. (2016); Soltau et al. (2016) and speech-2-text translation Chung et al. (2019) have shown promising results. Recently Haque et al. (2019) propose a method for learning audio-linguistuc embedding but that too depends on using transcribed text.

Due to the nature of the speech processing pipeline, natural language understanding tasks suffer from two major problems, 1) error propagation through ASR leading to noisy lexical features 2) loss of rich information which supplement lexical features, such as prosodic and acoustic expressive speech patterns.

In this chapter, we propose a framework to address the problem of predicting behavior codes directly from speech utterances. We focus on data from Motivational Interviewing (MI) sessions, a type of talk-based psychotherapy focused on behavior change. In psychology research and clinical practice, behavioral coding is often used to understand process mechanisms and therapy efficacy and outcomes. Behavior codes are annotated by an expert at an utterance level (or interaction level) by listening to the session. Examples of utterance level behavior codes include if there was a simple of complex reflection by the therapist of their patient's previous utterance(s). Several approaches have been proposed for automatic prediction of behavior codes, mainly using lexical features and/or linguistic features such as information from dependency trees Xiao et al. (2016a); Tanana et al. (2016); Pérez-Rosas et al. (2017); Cao et al. (2019); Gibson et al. (2019). Recent works Singla et al. (2018); Chen et al. (2019) reveal that using acoustic and prosodic features in addition to lexical features outperforms single modality models.

Speech2Vec Chung & Glass (2018) show that high quality word representations can be learnt by just using speech features. They learn word representations in an unsupervised manner using an objective similar to the Skipgram objective of Word2Vec Mikolov et al. (2013) (a word representation should be representative of its context words) and sequence-to-sequence framework. However, Speech2Vec only aims to learn word representations which are averaged spoken-word representations of that word in the corpus. Our proposed approach aims to exploit speech signal to word encoder learnt using an architecture similar to Speech2Vec as lower level dynamic word representations for the utterance classifier. Thus, our system never actually needs to know what word it is but only word segmentation information. We hypothesize word segmentation information can be obtained by cheaper tools, e.g. a supervised word segmentation system Tsiartas et al. (2009) or a heuristics based system based on acoustic and prosodic cues Junqua et al. (1994); Iwano & Hirose (1999). We plan to investigate the effect of noise in word boundaries on encoder quality in the future.

Our end-2-end transcription-free approach is similar and perhaps even motivated some of the previous works. There have been some works Serdyuk et al. (2018); Lugosch et al. (2019) which perform prediction tasks directly from speech signals but lack in capturing the underlying linguistic structure of a language (sentences break into words for semantics). We believe capturing some of the important linguistic units (e.g. words) are important for spoken language understanding. Qian et al. (2017) is most similar to our work in terms of overall architecture as they also first get word level representations and then use the encoder for utterance level prediction. However Qian et al. (2017) uses transcribed word transcriptions but we only use word boundaries for ASR-free end-2-end spoken language understanding. As shown in Figure 1, mosk previous works follow the upper pipeline. They start with a transcript (manually generated or through an ASR), which is first segmented into utterances. They then use word-embeddings for each word in the transcript before feeding it into a classifier to predict target behavior codes.

Our approach shows competitive results when compared to state-of-the-art models which use transcribed text. Our target application domain in this work is psychotherapy. While utterance level behavior coding is a valuable resource for psychotherapy process research, it is also a labor intensive task for manual annotation. Our proposed method which does not rely on transcripts should help with cheaper and faster behavioral annotation. We believe this framework can be a promising direction to directly perform classification tasks given a spoken utterance.

3.2 Our Approach

We first learn a word-level speech signal to word encoder using a sequence-to-sequence framework. Speech-2-Vector follows the learning objective similar to Skipgram architecture of Word2Vec. We then use the pre-trained encoder to predict behavior codes.

3.2.1 Speech signal to word encoder

Our Speech signal to word encoder (SSWE) encoder is an adaptation of Speech2Vec Chung & Glass (2018) which in turn is motivated by Word2Vec's skipgram architecture. The model learns to predict context words given a word. But unlike Word2Vec, in SSWE, each word is represented by a sequence of speech frames. We adopt the widely known sequence-to-sequence architecture to generate context words given a spoken word. Our model generates speech features for context words $(X_{n-4}, X_{n-3}, ..., X_{n+4})$ given speech features for a word X_n. As input for word X_n, it takes $K * 13$ dimensional MFCC features extracted from every 25 ms window of speech audio using a frame rate of 10ms. K is the maximum number of frames a spoken word can have. This input is then processed through a bidirectional LSTM layer Hochreiter & Schmidhuber (1997) to generate the context vector C. C is then used by a unidirectional LSTM decoder to generate the speech features for words in context $(Y_{n-4}, Y_{n-3}, ..., Y_{n+4})$. We optimize the model by minimizing the mean squared loss between predicted and target outputs: $\sum_{i=1}^{k} \|X^i - Y^i\|^2$. Following this approach, our system never uses any form of explicit transcriptions for learning the encoder, just only the word boundaries. Figure 3.2 gives a pictorial description of this process.

Our Speech-2-Vector encoder is trained using a speech corpus and word segmentation information. In our setup, we assume we have high quality word segmentation information. For the purpose of our experiments, we obtain the word segmentation information using a Forced-aligner Ochshorn & Hawkins (2016). The forced aligner primarily gives boundaries for the start and end of a word, which are then used to get speech features for a word. We hypothesize that learning word segmentation is a cheaper task than training a full-blown ASR.

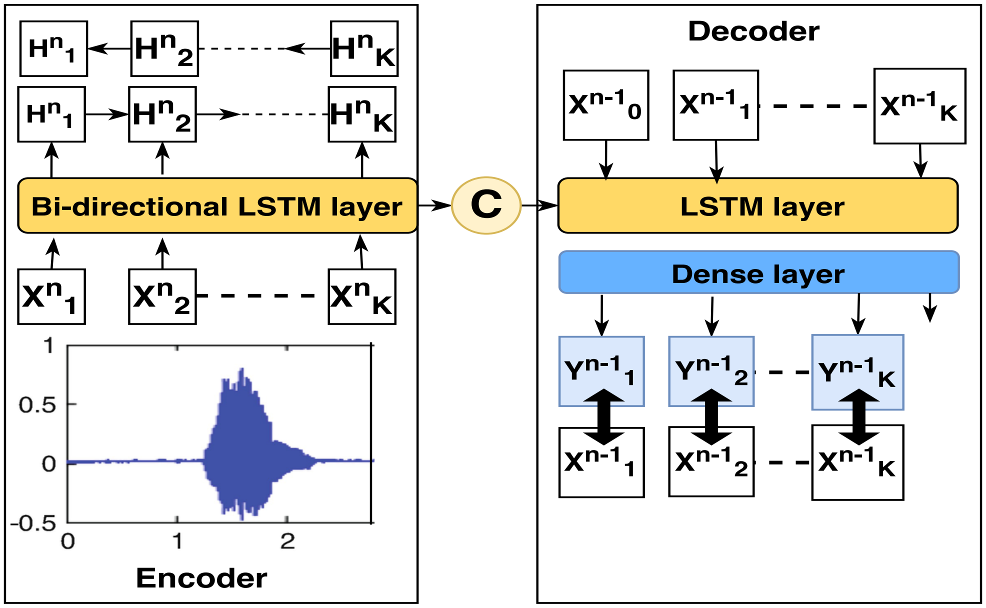


Figure 3.2: Speech signal to word encoder (SSWE) which uses sequence-2-sequence framework for generating representations of context words given a word.

3.2.2 Utterance classifier

Figure 3.3 shows the picturesque view of our utterance classifier. Given a word-segmented utterance, we first process speech features for each word to get word-level representations (W_i..... W_n). We then learn a function c = f(W) that maps W to a behavioral code $c1, 2, ..., C$, with C being the number of defined target code types.

We use a parametric composition model to construct utterance-level embeddings from word-level embeddings. Word-level representations ($W_i, ..., W_n$) are then fed into a bidirectional LSTM layer to contextualize the word embeddings. Contextualized word embeddings are then fed to a self-attention layer to get a sentence representation S which is then used to predict the behavior code for an utterance using a dense layer which projects it to C dimensions using a softmax

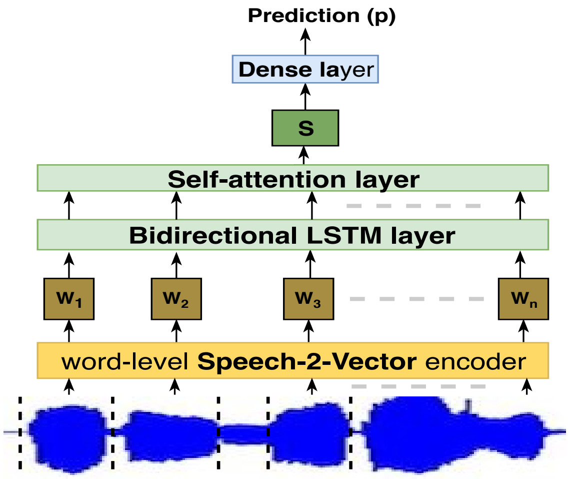


Figure 3.3: Classifier to predict behavior codes which takes input a word segmented speech signal and also uses pretrained Speech-2-Vector encoder to get word level representations.

operation. We use a self-attention mechanism similar to the one proposed in Yang et al. (2016)

3.3 Dataset

We experiment with two datasets for training the S2V encoder: first on the LibreSpeech Corpus Panayotov et al. (2015) (500 hour subset of broadband speech produced by 1,252 speakers) and second, directly on our classifier training data, which we describe below.

For classification, we use data from Motivational Interviewing sessions (a type of talk based psychotherapy) for addiction treatment presented in Tanana et al. (2016); Pérez-Rosas et al. (2017). There are 337 transcribed sessions (approx. 160 hours of audio) coded by experts at the utterance level with behavioral labels following the Motivational Interviewing Skill Code (MISC) manual Miller et al. (2003).Each human coder segmented talk turns into utterances (i.e., complete thoughts) and

Code	**Description**	**#Train**	**#Test**
FA	Facilitate	1194	496
GI	Giving information	12241	4643
RES	Simple reflection	4594	1902
REC	Complex reflection	3613	1235
QUC	Closed question (Yes/No)	4393	2066
QUO	Open question (Wh-type)	3871	1445
MIA	MI adherent	2948	1521
MIN	MI non-adherent	890	433
Total		33744	13741

Table 3.1: Data statistics for Behavior code prediction in Motivational Interviewing Psychotherapy

assigned one code per utterance for all utterances in a session. The majority of sessions were coded once by one of three expert coders.

In this chapter, we use the strategy proposed by Xiao et al. (2016a) grouping all counselor codes into 8 categories (described in Table 3.1). We remove backchannels without timestamps which cannot be aligned and split the data into training and testing sets by sessions with roughly 2:1 ratio. This split is consistent with all compared works.

3.4 Training details

Speech-2-Vector Encoder: We implemented the model with PyTorch Paszke et al. (2017). Similar to Chung & Glass (2018), we also adopted the attention mechanism which enables the Decoder to condition every decoding step on the last hidden state of the Encoder Subramanian et al. (2018). The window size was set to 4. We train the model using stochastic gradient descent (SGD) with learning rate of $1e*-3$ and batch size of 64 (spoken-word, context) pairs. We experimented with hyperparameter combinations for: using bidirectional or unidirectional RNNs, using GRU vs LSTM cell, number of LSTM hidden layers and learning rates. We found there was not a big difference in encoder output quality with higher dimensions. Therefore, we use a 50 dimensional LSTM cell, thus the resulting encoder output becomes 100 (Bidirectional last hidden states) + 100 (cell state) = 200 dimensions.

Utterance Classifier: The chosen batch size was 40 utterances. The LSTM hidden state dimension is 50. We use dropout at the embedding layer with drop probability 0.3. The dense layer is of 100 dimensions. The model is trained using the Adam optimizer Kingma & Ba (2014) with a learning rate of 0.001 and an exponential decay of 0.98 after 10K steps (1 step = 40 utterances). Similar to prior work, we also weight each sample according to normalized inverse frequency ratio.

3.5 Experiments & Results

Speech2Vec vs Word2Vec: Table 3.2 shows results where we compare performance of the system when we use lexically-derived word embeddings (word2Vec) vs speech-features derived word embeddings (Speech2Vec). If a word appears in a corpus n times, then speech2vec uses a system similar to our Speech-2-Vector encoder and averages them to get a word embedding for that dictionary word. Results confirm two main observations: 1) It is better to learn/fine-tune the word embeddings on an in-domain dataset. 2) Speech2Vec that learns word embeddings based on different spoken variations of word provides better results for behavior code prediction. This result is consistent with findings from Singla et al. (2018); Chen et al. (2019) where it is shown that acoustic-prosodic information can provide complementary information for predicting behavior codes and hence, produce better results. One challenge is that SSWE and Speech2Vec generally needs large amount of transcribed data to learn high quality word embeddings. Therefore, we first train

Model	**Word embeddings Data**	**F1-score**
Word2Vec[†]	Google-wiki	0.53
Word2Vec[†]	Indomain	0.56
Speech2Vec[†]	LibreSpeech	0.58
Speech2Vec[†]	Libre+Indomain*	**0.60**

Table 3.2: Using word embeddings learnt using speech features (Speech2vec) vs Word2Vec. * marks that model was only fine tuned for in-domain data. [†] marks that all these classifiers were trained end-2-end

SSWE on a general speech corpus (here, LibreSpeech (Libre)) before fine-tuning it on our classifier training data (results with $*$ show this experiment).

Transcriptions vs. No Transcriptions: Methods discussed above still rely on transcriptions to know what the word is. However, our proposed method does not use any explicit transcription but only the word segmentation information. Results in Table 3.3 show that using a pre-trained Speech-2-Vector encoder as a building block to get word representations can lead to competitive results to other methods which rely heavily on first generating transcripts of the spoken utterance. Here we also compare our model to the multimodal approach proposed by Singla et al. (2018); Chen et al. (2019) where they use word-level prosodic features along with lexical word embeddings. *Prosodic* and *Word2Vec+Prosodic*[†] show results for this system.

Table 3.3 also shows that doing end-2-end training (results with *) where our Speech-2-Vector encoder is also updated by the classifier loss generates poor results. We hypothesize that it can be due to the fact that our behavior code prediction data was split to minimize the speaker overlap. Thus it becomes easier to overfit when we fine-tune it on some speaker-related properties instead of generalizing for behaviour code prediction task.

Model	Pretrain data	F1-score
Majority class	-	0.33
Single-modality		
Word2Vec†	Indomain	0.56
Prosodic	Indomain	0.42
Multimodal		
Word2Vec+Prosodic†	Indomain	0.58
Speech2Vec†	Libre+Indomain*	**0.60**
Speech-only (Our approach)		
SSWE	Indomain	0.49
SSWE†	Indomain	0.44
SSWE	Libre+Indomain*	**0.56**
SSWE†	Libre+Indomain*	0.50

Table 3.3: We compare our proposed approach to previous approaches. Results in red are for the systems that do not use any transcriptions, only word segmentation information.

3.6 Conclusions

We show that comparable results can be achieved for behavior code prediction by just using speech features and without any ASR or human transcriptions. Our approach still depends on word segmentation information, however, we believe obtaining word segmentation detection system from speech is comparatively easier than building a high quality ASR. The evaluation results show the application significance of an end-2-end speech to behavioral coding for psychotherapy conversations. This allows for building systems that do not include explicit transcriptions, an attractive option for privacy reasons, when the end goal (as determined by the behavioral codes) is to characterize the overall quality of the clinical encounter for training or quality assurance.

3.7 Future work

The results still vary and are worse compared to human annotations. We plan to do a detailed analysis along two lines: 1) Comparing if proposed modeling technique can help bridge gap between predicted and human annotations, and 2) Effect of environment variable effect speech e.g. background noise, speaker features, different languages etc. We believe our approach can benefit from some straightforward modifications to the architecture, such as using convolutional neural networks which have shown to perform better at handling time-continuous data like speech.

3.8 Acknowledgements

This work was supported by the NIH.

Chapter 4

Leveraging Monolingual Text for Multilingual Text Representations

We present a novel multi-task modeling approach to learning multilingual distributed representations of text. Our system learns word and sentence embeddings jointly by training a multilingual skip-gram model together with a cross-lingual sentence similarity model. We construct sentence embeddings by processing word embeddings with an LSTM and taking a average of the outputs. Our architecture can transparently use both monolingual and sentence aligned bilingual corpora to learn multilingual embeddings, thus covering a vocabulary significantly larger than the vocabulary of the bilingual corpora alone. Our model shows competitive performance in a standard cross-lingual document classification task. We also show the effectiveness of our method in a limited resource scenario.

4.1 Introduction

Learning distributed representations of text, whether it be at the level of words, phrases, sentences or documents has been one of the most widely researched subjects in natural language processing in recent years (Mikolov et al., 2013; Pennington et al., 2014; Gouws et al., 2015; Socher et al., 2010; Pham et al., 2015; Kiros et al., 2015; Conneau et al., 2017; Le & Mikolov, 2014; Chen, 2017; Wu et al., 2017a). Word/sentence/document embeddings, as they are now commonly referred to, have quickly become essential ingredients of larger and more complex NLP systems looking to leverage the rich semantic and linguistic information present in distributed representations (Bengio et al., 2003; Maas et al., 2011; Collobert et al., 2011; Bahdanau et al., 2014; Chen & Manning, 2014).

Research that has been taking place in the context of distributed text representations is learning multilingual text representations shared across languages (Faruqui & Dyer, 2014; Bengio & Corrado, 2015; Luong et al., 2015). Multilingual embeddings open up the possibility of transferring knowledge across languages and building complex systems even for languages with limited amount of supervised resources Ammar et al. (2016); Johnson et al. (2016). By far the most popular approach to learning multilingual embeddings is to train a multilingual word embedding model that is then used to derive representations for sentences and documents by composition (Hermann & Blunsom, 2014). These models are typically trained solely on word or sentence aligned corpora and the composition models are usually simple predefined functions like averages over word embeddings (Lauly et al., 2014; Hermann & Blunsom, 2014; Mogadala & Rettinger, 2016) or parametric composition models learned along with the word embeddings (Schwenk et al., 2017). For a thorough survey of cross-lingual text embedding models, please refer to (Ruder, 2017b).

In this work we learn word and sentence embeddings jointly by training a multilingual skip-gram model together with a cross-lingual sentence similarity model. Our multilingual skip-gram model is similar to Luong et al. (2015). It transparently consumes *(word, context)* pairs constructed from monolingual as well as sentence aligned bilingual corpora. We process word embeddings with a bidirectional LSTM and then take an average of the LSTM outputs, which can be viewed as context dependent word embeddings, to produce sentence embeddings.

Since our multilingual skip-gram and cross-lingual sentence similarity models are trained jointly, they can inform each other through the shared word embedding layer and promote the compositionality of learned word embeddings at training time. Further, the gradients flowing back from the sentence similarity model can affect the embeddings learned for words outside the vocabulary of the parallel corpora. We hypothesize these two aspects of approach lead to more robust sentence embeddings.

The main motivation behind our approach is to learn high quality multilingual sentence and document embeddings in the low resource scenario where parallel corpus sizes are limited. The main novelty of our approach is the joint training of multilingual skip-gram and cross-lingual sentence similarity objectives with a shared word embedding layer which allows the gradients from the sentence similarity task to affect the embeddings learned for words outside the vocabulary of the parallel corpora. By jointly training these two objectives, we can transparently use monolingual and parallel data for learning multilingual sentence embeddings. Using a BiLSTM layer to contextualize word embeddings prior to averaging is orthogonal to the joint multi-task learning idea. We observed that this additional layer is beneficial in most settings and this is consistent with the observations of recent works on learning sentence and document embeddings such as Conneau et al. (2017); Yang et al. (2016)

4.2 Related Work

Cross-lingual Word Embeddings : Most approaches fall into one of these 4 categories: 1. monolingual mapping: learning transformations from other langauges to English Faruqui & Dyer (2014); Xing et al. (2015); Barone (2016), 2. pseudo cross-lingual: making a pseudo cross-lingual model and training off the shelf word embedding models Xiao & Guo (2014); Duong et al. (2016); Vulić & Moens (2016), 3. cross-lingual: learning embeddings using parallel corpora Hermann & Blunsom (2013); Chandar et al. (2014); Søgaard et al. (2015) and 4. joint optimization: using both parallel and monolingual corpora Klementiev et al. (2012); Luong et al. (2015); Vyas & Carpuat (2016); Coulmance et al. (2016). We adopt the skip-gram architecture of Luong et al. (2015) and train a single multilingual model using monolingual data from each language as well as any sentence aligned bilingual data available for any language pair.

Cross-lingual Sentence Embeddings: Some cross-lingual word embedding works also considered the problem of constructing sentence embeddings Vulic & Moens (2015); Pham et al. (2015); Hermann & Blunsom (2014). In general, it is not trivial to construct cross-lingual sentence embeddings by composing word embeddings as the semantics of a sentence is a complex language-dependent function of its component words as well as their ordering. Pham et al. (2015) addresses this difficulty by extending the paragraph vector model of Le & Mikolov (2014) to the bilingual context which models the sentence embedding as a separate context vector used for predicting the n-grams from both sides of the parallel sentence pair. At test time, the sentence vector is randomly initialized and trained as part of an otherwise fixed model to predict the n-grams of the given sentence. Our sentence embedding model is closer to the approach taken in Hermann & Blunsom (2014). They construct sentence embeddings by taking average of word or bi-gram embeddings and use a noise-contrastive loss based on euclidean distance between parallel sentence embeddings to learn these embeddings.

Multi-task Learning: Multi-task learning has been employed in various NLP applications where the parameters are shared among tasks Collobert & Weston (2008); Liu et al. (2016); Hashimoto et al. (2016). Liu et al. liu2016recurrent show the effectiveness of multi-task learning in multiple sentiment classification tasks by sharing an RNN layer across tasks while learning separate prediction layers for each task. Our multi-task architecture is unique in the sense that we treat training word embeddings as a separate task with a separate objective as opposed to pre-training them or training them only as part of a larger model.

4.3 Model

Our model jointly optimizes multilingual skip-gram Luong et al. (2015) and cross-lingual sentence similarity objectives using a shared word embedding layer in an end-to-end fashion.

Multilingual Skip-gram: Multilingual skip-gram model Luong et al. (2015) extends the traditional skip-gram model by predicting words from both the monolingual and the cross-lingual context. The monolingual context consists of words neighboring a given word as in the case of the traditional skip-gram model. The cross-lingual context, on the other hand, consists of words neighboring the target word aligned with a given source word in a parallel sentence pair. Figure 4.1 shows an example alignment, where an aligned pair of words are attached to both their monolingual and bilingual contexts. For a pair of languages $L1$ and $L2$, the word embeddings are learned by optimizing the traditional skip-gram objective with *(word, context word)* pairs sampled from monolingual neighbors in $L1 \rightarrow L1$ and $L2 \rightarrow L2$ directions as well as cross-lingual neighbors in $L1 \rightarrow L2$ and $L2 \rightarrow L1$ directions. In our setup, cross-lingual pairs are sampled from parallel corpora while monolingual pairs are sampled from both parallel and monolingual corpora.

Cross-lingual Sentence Similarity: We process word embeddings with a bi-directional LSTM Hochreiter et al. (2001); Hochreiter & Schmidhuber (1997)

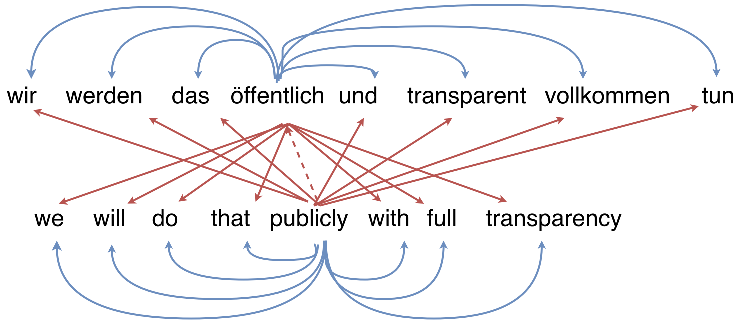


Figure 4.1: Example context attachments for a bilingual (en-de) skip-gram model.

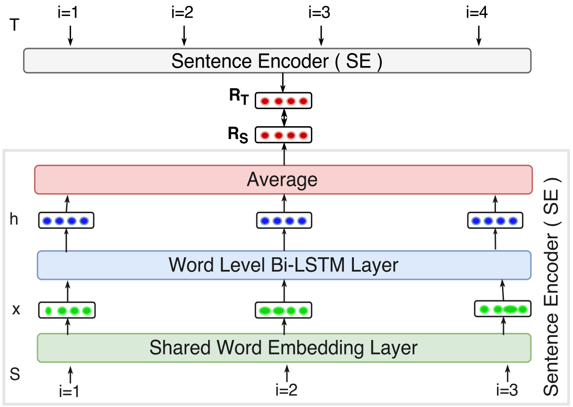


Figure 4.2: Overview of the architecture that we use for computing sentence representations R_S and R_T for input word sequences S and T.

and then take an average of the LSTM outputs (Figure 4.2). There are various implementations of LSTMs available; in this work we use an implementation based on Zaremba et al. (2014). The LSTM outputs (hidden states) contextualize input word embeddings by encoding the history of each word into its representation. We hypothesize that this is better than averaging word embeddings as sentences generally have complex semantic structure and two sentences with different meanings can have exactly the same words. Let $R : S \to \mathbb{R}_d$ denote our sentence encoder mapping a given sequence of words S to a continuous vector in $\mathbb{R}_d$. Given a pair of parallel sentences (S, T), we define their distance as $d(S, T) = \|R_S - R_T\|^2$. For every parallel sentence pair, we randomly sample k negative sentences $\{N_i | i = 1 \ldots k\}$ and define the cross-lingual sentence similarity loss as follows:

$$l(S, T) = \sum_{i=1}^{k} \max(0, m + d(S, T) - d(S, N_i))$$

Without the LSTM layer, this loss is similar to the BiCVM loss Hermann & Blunsom (2014) except that we use also the reversed sample (T, S) to train the model, therefore showing each pair of sentences to the model two times per epoch.

4.4 Experiments

4.4.1 Corpora

We learn the distributed representations on the Europarl corpus v71 Koehn (2005). For a fair comparison with literature, we use the first 500K parallel sentences for each of the English-German (en-de), English-Spanish (en-es) and English-French (en-fr) language pairs. We keep the first 90% for training and the remaining 10% for development purposes. We also use additional 500K monolingual sentences from the Europarl corpus for each language. These sentences do not overlap with the sentences in parallel data.

Words that occur less than 5 times are replaced with the <unk> symbol. In the joint multi-task setting, the words are counted in the combined monolingual and parallel corpora. The vocabulary sizes for German (de) and English (en) are respectively 39K and 21K in the parallel corpus, 120K and 68K in the combined corpus.

We evaluate our models on the RCV1/RCV2 cross-lingual document classification task Klementiev et al. (2012), where for each language we use 1K documents for training and 5K documents for testing.

4.4.2 Models

In addition to the proposed joint multi-task (JMT) model, **JMT-Sent-LSTM**, we also present ablation experiments where we omit the LSTM layer, the multilingual skip-gram objective or both. **JMT-Sent-Avg** is like the proposed model but does not include an LSTM layer. **Sent-LSTM** and **Sent-Avg** are the single-task variants of these models.

We construct document embeddings by averaging sentence representations produced by a trained sentence encoder. For a language pair $L1$-$L2$, a document classifier (single layer average perceptron) is trained on documents from $L1$, and tested on documents from $L2$. Due to lack of supervision on the $L2$ side, this setup relies on documents from different languages with similar meaning having similar representations.

4.4.3 Training

The single-task models are trained with the cross-lingual sentence similarity objective end-to-end using parallel data only. We also tried training word embeddings beforehand on parallel and mono data and tuning them on the cross-lingual sentence similarity task but that did not improve the results. Those results are omitted for brevity. The multi-task models are trained by alternating between the two tasks.

Multilingual Skip-gram: We use stochastic gradient descent with a learning rate of 0.01 and exponential decay of 0.98 after 10K steps (1 step is 256 word pairs), negative sampling with 512 samples, skip-gram context window of size 5. Reducing the learning rate of the skip-gram model helps in the multi-task scenario by allowing skip-gram objective to converge in parallel with the sentence similarity objective. At every step, we sample equal number of monolingual and cross-lingual word pairs to make a mini-batch.

Cross-lingual Sentence Similarity: The batch size is 50 sentence pairs. LSTM hidden state dimension is 128 or 512. We use dropout at the embedding layer with drop probability 0.3. Hinge-loss margin m is equal to sentence embedding size. We sample 10 negative samples for the noise-contrastive loss. The model is trained using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.001 and an exponential decay of 0.98 after 10K steps (1 step is 50 sentence pairs).

4.4.4 Results

Table 4.1 shows the results for our models and compares them to some state-of-the-art approaches. When the sentence embedding dimension is 512, our results are close to the best results from literature. When the sentence embedding dimension is 128, our JMT-Sent-LSTM model outperforms all of the systems compared. Models with an LSTM layer (Sent-LSTM and JMT-Sent-LSTM) perform better than those without one. Joint multi-task training consistently improves the performance. The results for the data ablation experiments (*no-mono) suggest that the gains obtained in the JMT setting are partly due to the addition of monolingual data and partly due to the multi-task objective.

Varying monolingual vs parallel data: The main motivation behind the multi-task architecture is to create high quality embeddings in the limited resource scenario. The bottom section of Table 4.1 shows the results for 128 dimensional embeddings when parallel data is limited to 100K sentences. JMT-Sent-LSTM

Model	en → de	de → en
500k parallel sentences, dim=128		
BiCVM-add+	86.4	74.7
BiCVM-bi+	86.1	79.0
BiSkip-UnsupAlign	88.9	77.4
Our Models		
Sent-Avg	88.2	80.0
JMT-Sent-Avg	88.5	80.5
Sent-LSTM	89.5	80.4
JMT-Sent-LSTM	**90.4**	**82.2**
JMT-Sent-Avg*no-mono	88.8	80.3
JMT-Sent-LSTM*no-mono	89.5	81.5
100k parallel sentences, dim=128		
Sent-Avg	81.6	75.2
JMT-Sent-Avg	85.3	79.1
Sent-LSTM	82.1	76.0
JMT-Sent-LSTM	87.4	80.7
JMT-Sent-LSTM*no-mono	83.4	76.5

Table 4.1: Results for models trained on en-de language pair. *no-mono means no monolingual data was used in training. We compare our models to: BiCVM-add+ Hermann & Blunsom (2014), BiCVM-bi+ Hermann & Blunsom (2014), BiSkip-UnsupAlign Luong et al. (2015) and para_doc Pham et al. (2015).

results in this scenario are comparable to the results from the middle section of Table 4.1 which use 500K parallel sentences. These findings suggest that JMT-Sent-LSTM model can produce high quality embeddings even with a limited amount of parallel data by exploting additional monolingual data. Table 4.2 compares Sent-LSTM vs. JMT-Sent-LSTM at different data conditions. JMT-Sent-LSTM produces consistently better embeddings as long as the amount of additional monolingual data is neither too large nor too small compared to the amount of parallel data – 3-4 times parallel data size seems to be a good heuristic for choosing monolingual data size.

Multilingual vs Bilingual models: Table 4.3 compares multilingual models (en, es, de) to bilingual models. First four rows of Table 4.3 show results for

Parallel Mono	20K	50K	100K	500K
no-mono	60.3	68.3	82.1	89.5
20K	57.4	68.7	80.2	89.5
50K	**62.7**	69.0	83.5	89.5
100K	61.5	71.9	85.1	89.6
200K	58.1	**72.1**	85.5	90.0
500K	52.6	64.8	**87.4**	**90.4**

Table 4.2: Sent-LSTM vs. JMT-Sent-LSTM at different data conditions (en-de, dim=128).

Model	en-es	en-de	de-en	es-en	es-de
Sent-Avg	49.8	86.8	78.4	63.5	69.4
Sent-LSTM	53.1	89.9	77.0	67.8	65.3
JMT-Sent-Avg	51.5	87.2	75.7	60.3	**72.6**
JMT-Sent-LSTM	**57.4**	**91.0**	75.1	63.3	68.1
JMT-Sent-LSTM*	54.1	90.4	**82.2**	**68.4**	-

Table 4.3: Multilingual vs. bilingual* models (dim=128).

multilingual systems where sentence encoder is trained for three languages (en,es,de) using en-es and en-de parallel data and additional monolingual data for each language. Document representations obtained from this sentence encoder are then used to train a classifier for a language pair like en-de, where the classifier is trained on en documents and then tested on de documents. In this scenario, we can build classifiers for language pairs like es-de even though we do not have access to es-de parallel data since embeddings we learn are shared between the three languages. Bottom row in Table 4.3 shows results for bilingual systems where we train the sentence encoder for two languages, and then use that encoder to train a document classifier for one language and test on the other. In this scenario, we cannot build classifiers for language pairs like es-de for which we do not have access to parallel data.

Multilingual models perform better than bilingual ones when English is the source language but they perform worse in the other direction. We believe this discrepancy is because Europarl documents were originally in English and later translated to other languages. The multilingual models also show promising results for es-de pair, for which there was no parallel data.

4.5 Linguistic analysis

As classification experiments focused on keeping semantic information in sentence level representations, we also checked if produced word embeddings still made sense. We use JMT-Sent-LSTM model for this purpose. Figure 4.3 shows t-SNE projections for some sample words. Even though the model didn't use any German-Spanish parallel data it managed to map words which have similar meaning (transkribiert and transcribió) closer. Words that are antonyms but still have a similar meaning are close to each other (cunnigly (en), honestly (en) and astucia (es)). Nearest neighbors in the multilingual representation space are generally of same form across languages. It can also be observed that English words lie towards the middle of Spanish and German words which we believe is due to English being the pivot for the other two languages.

4.6 Conclusion

Our results suggest that joint multi-task learning of multilingual word and sentence embeddings is a promising direction. We believe that our sentence embedding model can be improved further with straightforward modifications to the sentence encoder architecture, for instance using stacked LSTMs or batch/layer normalization, and addition of sentence level auxiliary tasks such as sentiment

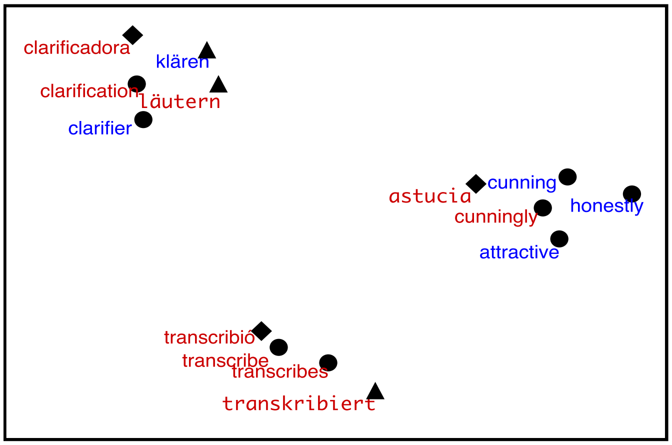


Figure 4.3: t-SNE projections for 3 English words (clarification, transcribe, cunningly) which are not in the parallel corpus and their four nearest neighbors. Red words are only in the monolingual corpus. Blue words exist in parallel corpus too.

classification or natural language inference. We plan to explore these directions and evaluate our approach on additional tasks in the future.

4.7 Discussion and Future Work

In our exploration of architectures for the sentence encoding model, we also tried using a self-attention layer following the intuition that not all words are equally important for the meaning of a sentence. However, we later realized that the cross lingual sentence similarity objective is at odds with what we want the attention layer to learn. When we used self attention instead of simple averaging of word embeddings, the attention layer learns to give the entire weight to a single word in both the source and the target language since that makes optimizing cross lingual sentence similarity objective easier. Another approach could be to derive high dimensional embeddings in a way similar to Conneau et al. (2017) and using max-pooling which can allow efficient selection for each dimension to represent meaning.

Even though they are related tasks, multilingual skip-gram and cross-lingual sentence similarity models are always in a conflict to modify the shared word embeddings according to their objectives. This conflict, to some extent, can be eased by careful choice of hyper-parameters. This dependency on hyper-parameters suggests that better hyper-parameters can lead to better results in the multi-task learning scenario. We have not yet tried a full sweep of the hyper-parameters of our current models but we believe there may be easy gains to be had from such a sweep especially in the multi-task learning scenario. Other thing that remains rather unexplored is to do other levels of multitasking, like learning character representations or multitasking at sentence level.

Chapter 5

Leveraging Monolingual text for Document Classification

This chapter describes a novel approach of multitask learning which uses end-to-end optimization technique for document classification. The application motivation comes from the need to extract "Situation Frames (SF)" from a document within the context of DARPA's LORELEI program targeting humanitarian assistance and disaster relief. We show the benefit of our approach for extracting SF: which includes extracting document types and then linking them to entities. We jointly train a hierarchical document classifier and an auto-encoder using a shared word-level bottleneck layers. Our architecture can exploit additional monolingual corpora in addition to labelled data for classification, thus helping it to generalize over a bigger vocabulary. We evaluate these approaches over standard datasets for this task. Our methods show improvements for both document type prediction and entity linking.

Multitask learning, text classification, situation awareness

5.1 Introduction

Multi-task learning (MTL), a widely used approach in NLP, comes in many guises: joint learning Hashimoto et al. (2016); Chidambaram et al. (2018); Arik et al. (2017), learning to learn Baxter (1997), and learning with auxiliary tasks Zhang et al. (2014); Liu et al. (2015); Gupta et al. (2017) are some of the names used for it. MTL has also helped achieve state-of-art results for wide range of NLP problems Hashimoto et al. (2016); Wu et al. (2017b); Liu et al. (2015); Søgaard & Goldberg (2016). Wu et al. (2017b) show jointly modeling the target word sequence and its dependency tree structure helps to improve dependency parsing results. Recently Chidambaram et al. (2018); Singla et al. (2018) show that multitask learning can also help transfer knowledge among tasks through shared lower level layers. For a thorough survey of multitask learning objectives in NLP, please refer to Ruder (2017a).

In this work, we jointly train two tasks, namely, *English auto-encoder* and a Hierarchical Attention based Document Classifier (HADC). HADC classifier, is similar to Yang et al. (2016) where given an input document it exploits the hierarchical structure of a document. It uses Bi-LSTMs and self-attention mechanism for encoding words to sentence embeddings, and then sentences to document embedding. Our English auto-encoder is similar to Artetxe & Schwenk (2018) but adapted to a monolingual scenario. It aims to contain most information about a sentence in contextualized word embeddings (i.e., output of word level Bi-directional LSTM layer which takes word embeddings as input). The HADC and English auto-encoder share the word level word embedding and Bi-LSTM layer. In the multitask setup, we train both these tasks jointly using a weighted loss. We hypothesize that due to joint training both these tasks can inform each other through shared layers, which enables HADC classifier to be trained in an end-to-end fashion on a bigger vocabulary than labelled data.

Our motivating application comes from DARPA LORELEI program which annotates and aims to make systems that can provide that Situation awareness results in form of Situation Frames (SF) Christianson et al. (2018), notably for humanitarian assistance and disaster relief. Given a speech recording or a text document (including social media), a system should predict SFs.

SF is defined by a *(Type, PlaceMention)* pair, where *Type* refers to a situation type and *PlaceMention* refers to a location. Linking of a situation *Type* to a *PlaceMention* is called *Localization* Cheung et al. (2017). There can be multiple frames in a document. For a system that can work across languages, people either use machine translation (MT) systems to go from a given language to English and then use an English-only system for SF prediction Mihalcea et al. (2007); Shi et al. (2010); Malandrakis et al. (2018); Wiesner et al. (2018); Cheung et al. (2017). Alternatively, one can use pre-trained fixed multilingual word embeddings as low level features directly for the SF system Guo & Xiao (2012); Xu et al. (2016); Muis et al. (2018). For speech sources, a widely followed approach is to transcribe the audio sessions using an ASR and then translate them into English using a Machine Translation system Malandrakis et al. (2017, 2018); Papadopoulos et al. (2017). Since a majority of the annotated SF resources are in English, therefore in this chapter, we focus on applying the concept of MTL learning for the task of predicting situation frames for English. We propose an end-to-end system which takes as input a pseudo-entity level document (using segments in which a *PlaceMention* appears) alongside the original source document, and predicts whether there should be a *Type* linked to this *PlaceMention* or not.

Our multitask models show improvements for F-score measure across languages (translated into English) for both *Type* and Localization prediction. We believe these improvements are due to the system's ability to avoid overfitting and generalize as our document classifier is trained in a multitask fashion with an English auto-encoder. Results also suggest that doing an end-to-end *localization* for extraction of situation frames gives better results.

Language	**En**	**En-twt**	**Mn**	**Ug**	**Tg**	**Or**	**Kn**	**Sn**	**Bn**	**Hi**	**Th**	**Zu**
Entities / Entity Documents	3152	-	1368	1120	1712	2484	627	566	746	624	682	975
Documents	756	2934	132	256	656	1264	339	331	144	162	158	408

Table 5.1: Frequency for documents and gold standard entities for each language from LDC.

5.2 Data

LDC[1] provides annotated data packages in multiple languages along with their translations. A situation Type is selected from the fixed inventory of eleven labels, namely, *Evacuation*, *Food-supply*, *Search/rescue*, *Utilities*, *Infrastructure*, *Medical-assistance*, *Shelter*, *Water supply*, Terrorism, *Crime-voilence* and *Regime-change*

We use all data available for following languages: English (En), Spanish (es), Mandarin (Mn), Ugyhur (Ug), Tagrinya (Tg), Oromo (Or), Bengali (Bn), Hindi (Hi), Thai (Th) and Zulu (zu), Kinyarwanda(Kn) and Sinhalese (Sn) along with their English translations and SF annotations. We also collect additional annotations for English tweets (En-twt) and assume that all *PlaceMentions* as linked to the *Type* of the tweet. Table 5.1 shows frequencies of collected data.

5.3 Multitask Learning

We propose a multitask architecture for the extraction of situation frames. The main idea is to jointly train a hierarchical attention document classifier and an English auto-encoder. Both these tasks share the word level variables i.e., word embeddings and word level Bi-LSTM layer (context dependent word embeddings)

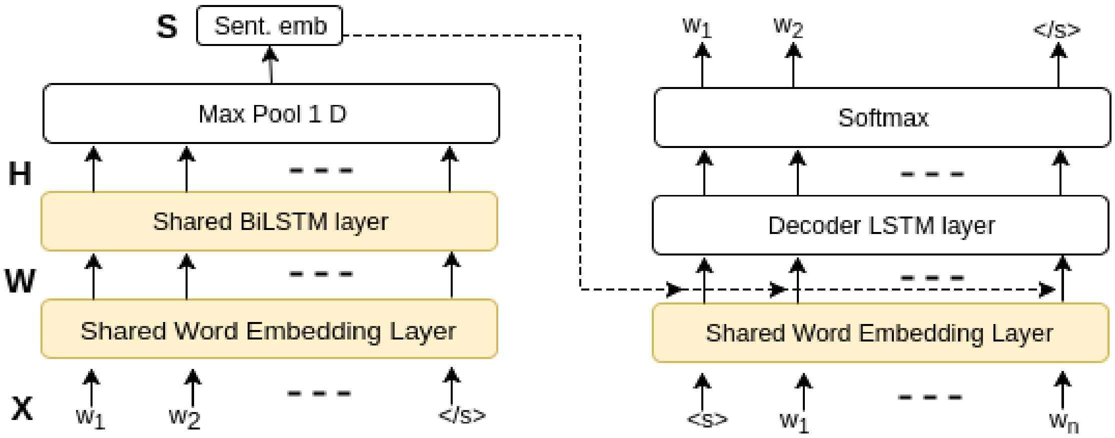


Figure 5.1: English Auto-encoder architecture to learn contextualized word embeddings. Boxes in yellow shows variables which are shared among tasks.

5.3.1 English Auto-encoder

Our auto-encoder uses contextualized word embeddings i.e., Bi-LSTM layer shared between two tasks. There are various implementations of LSTMs available; in this work we use an implementation based on Zaremba et al. (2014) which comes as a part of Tensorflow Abadi et al. (2016). Sentence embeddings are obtained by applying a max-pooling operation over the output of a word level Bi-LSTM layers. Similar to Artetxe & Schwenk (2018) these sentence embeddings are also used to initialize the decoder LSTM through a linear transformation, and are also concatenated to its input embeddings at every time step. As shown in Figure 5.1, there is no other connection between the encoder and the decoder as we want contextualized word embeddings to capture all the information of a sentence. This is done because contextualized word embeddings (output of word-level Bi-LSTM layers) is shared with the Document classifier described in the following section.

[1] https://www.ldc.upenn.edu/

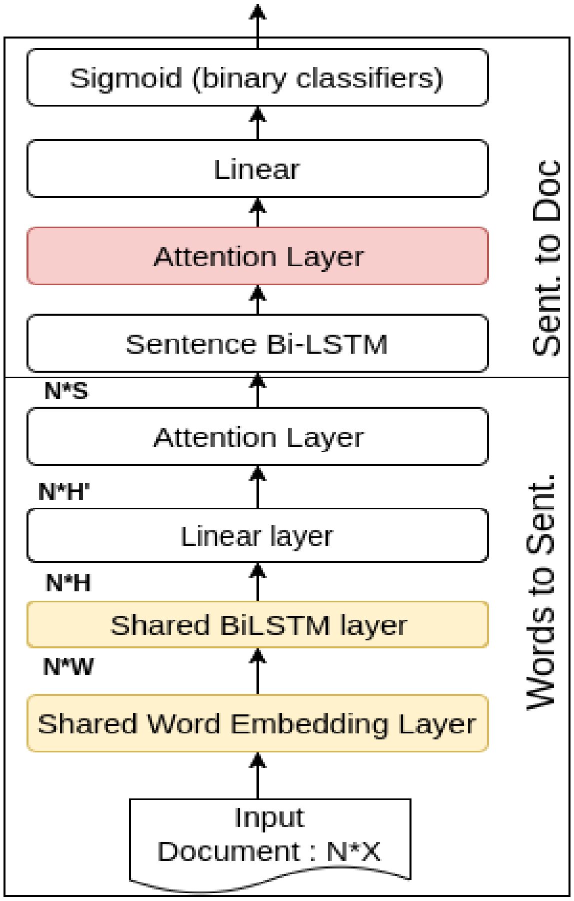


Figure 5.2: Hierarchical Attention based Document Classifier (HADC) architecture which shares lower level layers with English Autoencoder.

5.3.2 Hierarchical Attention based Document Classifier (HADC)

Our HADC architecture (Figure 5.2) is similar to Yang et al. (2016), where given an input sentence X, we first feed word embeddings W to a Bi-LSTM layer to get contextualized word embeddings H. These contextualized word embeddings are then passed through a dense layer to get H'. H' is then passed to a self-attention layer to get a representation for each sentence S in the document. We use an attention layer (equations 1-3) with an internal context vector (Yang et al., 2016).

$$k_i = \tanh(WH'_i + b) \tag{5.1}$$

$$\alpha_i = \text{softmax}(k_i^T a) \tag{5.2}$$

$$S = \sum_i \alpha_i h_i \tag{5.3}$$

The attention layer first applies a one-layer MLP to its inputs H'_i to derive the keys k_i. Then it computes the attention weights by applying a softmax non-linearity to the inner products between the keys k_i and the internal context vector a . Finally it computes the sentence representation R by taking a weighted average of its inputs. The context vector a is a model parameter that is initialized with uniform weights so that it behaves like an averaging operation at the beginning of the training.

For going from sentences to documents, we follow the same architecture again. i.e., keeping a Bi-LSTM layer to contextualize sentence embeddings and then an attention layer to get a document representation. This document representation is then fed to task specific linear and sigmoid layer. We use the sigmoid layer because our tasks are binary multilabel.

Multitask Loss: Our final multitask loss function is made of two terms, auto-encoder reconstruction loss A and HADC classification loss B. The *Loss* is defined as $B + \alpha * A$, where α is empirically set to 0.3 for all multitask experiments.

5.3.3 Training Routine

We remove all punctuation and lower-case the data. Words that occur less than 5 times are replaced with the <unk> symbol. We initialize the word emdedding layer using 300 dimensional pre-trained Word2Vec embeddings[2]. We use batch

[2] https://code.google.com/archive/p/word2vec/

size of 50 sentences for auto-encoder and 20 documents for HADC. An additional 300 random sentences from HADC mini-batch are also added to the auto-encoder mini-batch. The LSTM hidden state dimension is 256 and 128 for word-level and sentence-level layers, respectively. We use dropout at the embedding layer and before the sentence-level layer in HADC with drop probability of 0.3. We use Adam optimizer Kingma & Ba (2014) with a learning rate of 0.001 and an exponential decay of 0.98 after 10K steps (1 step is 1 mini-batch). The auto-encoder is pre-trained for 30K steps, before we begin joint multi-task training for document classification. We realized this pre-training helps simultaneous convergence of both the tasks.

Similar to prior works Mihalcea et al. (2007); Shi et al. (2010); Malandrakis et al. (2018); Wiesner et al. (2018); Cheung et al. (2017), we use the ReliefWeb corpus[3] of disaster-related documents to pre-train the model. The corpus contains disaster-related documents from various sources annotated for theme and disaster type, where theme labels are similar to topics discussed (food, water). We get inventory of 40 categories for the task of multi-label classification of documents and use approximately 120K documents for training and 52K documents for testing. We keep another 10K documents for the validation set. Our multitask model shows slightly better performance for ReliefWeb type prediction task. We achieve F-score of 72.5 and 73.1 for HADC and jointly trained multi-task HADC model respectively.

We have three different SF models:

- **HADC:** Pre-trained Relief web model is fine-tuned for situation type prediction by replacing last sigmoid layer.

[3] ReliefWeb website. http://reliefweb.int/. Retrieved 31 Mar 2016

- **HADC SepATT:** Here we use a different randomly uniform initialized internal context vector for sentence-level attention layer while predicting situation frames.

- **HADC Multi:** The English auto-encoder is pre-trained for 30K steps, before we start joint multitask training along with HADC SepATT model.

- **HADC Multi*extra:** Same as HADC-Multi, but here we use additional 500K sentences from EuroparlKoehn (2005) corpus for training English autoencoder

5.4 Localization

An important aspect of extracting a SF is linking a situation type to a PlaceMention. For this we build upon the approach from Malandrakis et al. (2018). It follows the hypothesis that segments in which an entity mention appears is predictive of the situation type. So to localize, they use a simple solution of creating location-specific sub-documents and attempt to classify them using the same models. For each detected *PlaceMention*, all sentences/segments that contain said *PlaceMention* are collected to form a dummy "document" per PlaceMention. These dummy documents are then passed through the SF model again, creating a set of *Type* labels per *PlaceMention*. The PlaceMention-level Types are filtered by the document-level Types: Types not detected during the document-level pass were not allowed at the entity level.

We follow the same hypothesis as used by Malandrakis et al. (2018) but instead of creating situation frames by just filtering out PlaceMention-level types using types predicted for a document, we provide posteriors of document level types as an input to predict PlaceMention-level type. We combine the loss of PlaceMention-level and document-level type prediction using a scaling parameter β and do a joint optimization. This allows our model to train end-2-end for predicting *Type* for a given *PlaceMention*. We performed experiments using various β values $(0.3, 0.5, 0.8)$ and found 0.5 gives best results for Localization.

5.5 Evaluation

We use English translations of Tigrinya (Tg), Oromo (Or), Kinyarwanda (Kn) and Sinhalese (Sn) for testing as they were official test languages for last two official LORELEI DARPA evaluations and remaining documents for training and validation (4950 documents and 11601 Entity-segments-documents using *PlaceMentions*). We randomly keep 10 % of data for validation.

System	**Tg**	**Or**	**Kn**	**Sn**
HADC	0.52	0.24	0.46	0.25
HADC SepATT	0.53	0.27	0.46	0.30
HADC Multi	0.50	**0.32**	0.51	0.41
HADC Multi*extra	**0.62**	0.31	**0.52**	**0.48**

Table 5.2: Situation Type F-scores using Human translations. *extra means 1M additional monolingual data from EUROPARL was used in training for English auto-encoder

System	**Tg**	**Or**	**Kn**	**Sn**
Nikos et al. Malandrakis et al. (2018)	0.21	0.08	0.20	0.13
HADC SepATT	0.24	0.12	0.24	0.19
HADC Multi	**0.27**	0.16	0.26	0.21
HADC Multi*extra	0.25	**0.20**	**0.30**	**0.25**

Table 5.3: Situation Type + Place (Gold Standard PlaceMentions) F-scores. *extra means 1M additional monolingual data from EUROPARL was used in training for English auto-encoder

Table 5.2 shows our results for F-score measure for the multi-label situation *Type* prediction at the document level. Our multi-task model *HADC-Multi* shows improvement over the baseline *HADC* model except for Tigrinya (Tg). This suggests multitasking helps achieve better results. The results for the experiments with additional monolingual data (*extra) shows best performance. This suggests that these improvements are mainly due to added vocabulary provided by additional monolingual data. With regards to doing *Localization*, we report f-score measures for the *(Type, PlaceMention)* tuple. Table 5.3 shows results for Localization. Similar to *Type* prediction results model with additional monolingual data (*extra) outperforms other compared models. All our end-to-end optimized models show gains when compared to the previous approach followed by Malandrakis et al. (2018).

5.6 Conclusions & Future work

Our results suggest that joint multi-task learning of contextualized word embeddings using an English auto-encoder and end-to-end optimization for extracting situation frames is a promising direction. We believe our multitask architecture can be improved further by straightforward modifications to the output of shared layers like applying domain adversarial penalty to the contextualized word embeddings. We intend to apply, adapt and test our architecture to other NLP and speech tasks in the future like sentiment classification and emotion recognition.

5.7 Acknowledgements

We would like to thank other ELISA team members who contributed to idea formulation, Dogan Can (USC) and Jonathan May (ISI). This work was supported by the U.S. DARPA LORELEI Program under Contract No. HR0011-15-C-0115 with the University of Southern California.

Chapter 6

Factored Automatic Extractive Summarization

Extractive summarization methods extract parts of text verbatim that are deemed informative to create a summary. However, existing extractive methods rely only lexical words for addressing the problem of redundancy and diversity of information. in this chapter, we propose extensions of widely used extractive alogorithms, namely: Luhn, MMR (Max marginal relevance) and Textrank which can incorporate information from not just lexical context but also other factors. Our results indicate that use of factors like psycholinguistics and unsupervised topic models can help to improve quality and provide better context for the summaries. Our results on a standard meeting corpus suggest that these factors help generate more relevant summaries. Our proposed extension while allows a user to control contribution of each information type improves over some of the well-known methods for both automatic and manual evaluation metrics.

abstractive summarization, psycholinguistics, text summarization

6.1 Introduction

From business meetings to therapy sessions, humans take part in many conversations that require note-taking to represent the information discussed in a more compact manner for future use. Over the years, many different methods have been proposed to do automatic text summarization (Mihalcea, 2004; Hong & Nenkova, 2014; Fein et al., 1999). These fall within two categories: abstractive and extractive summarization. Abstractive summarization aims to extract a low-level representation of a document or a conversation and uses a natural language generation system to develop summaries (Paulus et al., 2017; Liu et al., 2018). Extractive summarization, which is also the topic of this chapter, aims to either extract important information in a greedy-fashion to either do slot-filling (Oya et al., 2014) or segment ranking/selection to make a summary (Alguliev et al., 2009; Pittaras & Karkaletsis, 2019).

Past research has focused heavily on summarizing long pieces of continuous text, like documents (Alguliev et al., 2009; Prasojo et al., 2018). However, there has been a growing interest in summarizing spoken conversations (Ganesh & Dingliwal, 2019). Some of the famous extractive methods that have been heavily explored, such as maximum marginal relevance (MMR) (Zhong et al., 2019) and LexRank (Ramesh & Rajan, 2019), decide the importance of each sentence in text and then take the most important sentences verbatim to create a summary of the original text either by retrieval or re-ranking. These methods attempt to reduce redundancy in generated summaries based on similar vocabulary in sentences (Alguliev et al., 2009). These simpler yet effective information compression based text summarization techniques don't require any form of supervised data. However this dependence on lexical raw words to address the issue of redundancy ignores the contribution of additional factors (e.g: indicators of affect, topic models or any other knowledge bases) which can lead to elimination of high representation of prolonged yet unimportant conversation segments. In existing form, they lack in exploiting higher-level factors to better aid the sentence ranking/extraction process for compression of information. We believe incorporating some of these factors can help in more informed, guided and user-controlled text summarization. Some recent works have explored incorporating semantic augmentation of distributed word representations passed into summarizers (**?**) and also augmenting text summarizers to be more receptive to speech summarization (**?**). However, there is limited research in underlying a framework for systematic incorporating multiple factors to do a wholesome and concise text compression.

In this work, we propose augmentations of three prominent extractive summarization algorithms (MMR, Luhn and TextRank) which can exploit external factors, in particular sentence-level annotations of *affect* (in form of psycholinguistic information) and unsupervised topic models. Our proposed factored text summarization methods exploit information from namely two sources: 1) LIWC dictionary (Pennebaker et al., 2001), which provide information about psycho-linguistics based on language use and 2) unsupervised topic modeling based on distributed sentence representations. For our experiments, we use equal weights for all factors, however they can be tuned to get the desired type of summary. We hypothesise that our proposed method allows us to create less redundant summaries with an effective amount of context and content that is more useful for future reference of a conversation. Figure 6.3 gives a high-level view of our approach in the context of MMR, where each utterance is provided a $Score_K$ using a weighted output of similarity scores between an utterance and already generated summaries using a distance metric.

Our hypothesis is that humans naturally emote when speaking about ideas that have been most importance to them. Thus, high-sentiment (negative or positive) and psycholinguistic state often correlate with the essential underlying cognitive information in a conversation. Recently some other works have shown effectiveness of incorporating sentiment-information in abstractive summarizationGerani et al.

(2014); Liu et al. (2018). Furthermore, we argue that many human-generated summaries contain a lot of noise along with important content which can be eliminated with the application of our proposed augmentations of psycholinguistics and unsupervised topic modeling to popular extractive summarization methods. We evaluate our proposed approach using the standard AMI meeting corpus. The generated summaries are able to put the important portions of a human-generated summary together more succinctly than both the original algorithms and the human-produced summaries. Both automatic and human evaluation results indicate that the summaries generated with the augmented extractive summarizers are more useful to the general public.

We describe our feature representation is the next section, followed by our proposed extensions to extractive algorithms. We then share our results and then conclude with a brief discussion.

6.2 Features Used for Abstractive Summarization

We leverage additional sentence-level features: sentence-level representation of psycholinguistics information and unsupervised topic modeling information. We believe these features help increase not only the generated summary's non-redundancy but also improve the overall quality of the extracted summaries in terms of covering wide-range of psycho-linguistic phenomenon and vivid information.

Extractive summarization generally works by first getting a representation of each sentence in the document. This representation is generally a term (word) frequency vector for a sentence based on each document. We propose that we

Category	Sample Words
Affect Words	happy, cried
Social Words	mate, talk, they
Cognitive Processes	cause, know, ought
Perceptual Processes	look, heard, feeling
Biological Processes	eat, blood, pain
Core Drives and Needs	friend, bully, doubt
Relativity	area, bend, exit

Table 6.1: Incorporated LIWC Categories and Related Words

represent each sentence using 3 information vectors based on 1) Term frequency, 2) LIWC psycholinguistic statistics 3) One-hot Topic representation.

We extract these sentence level representations to incorporate two factors: psycholinguistic statistics and unsupervised topic modeling.

6.2.1 Linguistic Inquiry and word count

LIWC (Pennebaker et al., 2001) is a text processing application that processes raw text and outputs percentage of words from the text that belong to linguistic, affective, perceptual and other dimensions. It operates by maintaining a diverse set of dictionaries of words each belonging to a unique dimension. The resulting language categories were created to capture people's social and psychological states (Hong & Nenkova, 2014), so its representation of text (in our case, sentences) can help in capturing additional meaning. In our work, we only use information from the 7 categories shown in Table 6.1 in addition to the swear words subcategory. These categories were chosen due to their relation with highly sentimental areas of diction. Each category is represented in the sentence-level *affect* vector via the count of each category word in the sentence. Figure 6.1 describes this process in brief.

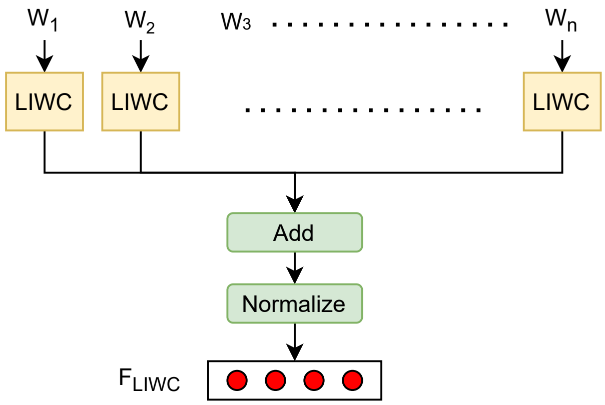


Figure 6.1: Representing a sentence based on LIWC dictionary. Every word is checked through a LIWC dictionary to check if that belongs to a category listed in Table 1. LIWC statistics for every word is then summed up and then normalized to get a sentence-level representation F_{LIWC}

6.2.2 Unsupervised Topic Models

In order to reduce redundancy and thus, keep the summary succinct, we use unsupervised topic models. This is useful as these topic models can be learn in an unsupervised manner on a bigger data. We believe this helps to ensure that utterances that are about an over-represented subject area in the transcript are limited in their representation in the generated summary. As this topic models are learnt using larger psychology corpus, the topics here are more general than represented by words within a document.

We obtain a distributed sentence representations using a sentence encoder. Sentence encoder in our case is just an average of distributed word representations which has been shown to be useful for variety of NLP tasks. Distributed word representations, also known as word embeddings are learnt using the standard Word2Vec (Lilleberg et al., 2015) skip-gram architecture. We learn 100 dimensional word embeddings for our corpus. We pretrain the system on Google-wiki corpus before we fine tune for our corpus. We then use the average of these word embeddings

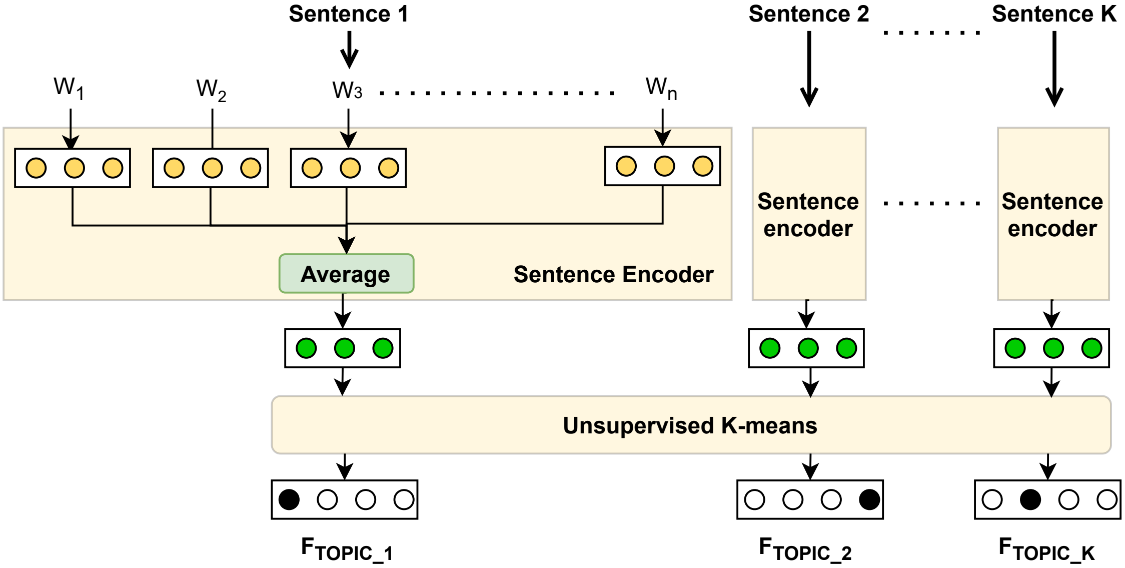


Figure 6.2: Representing a sentence based on active K-means topic models. Every sentence is first passed through a Sentence encoder. Then sentence representations are passed to K-means algorithm for clustering. F_{Topic_1} is obtained by one-hot representations of the topic incoming sentence belongs to.

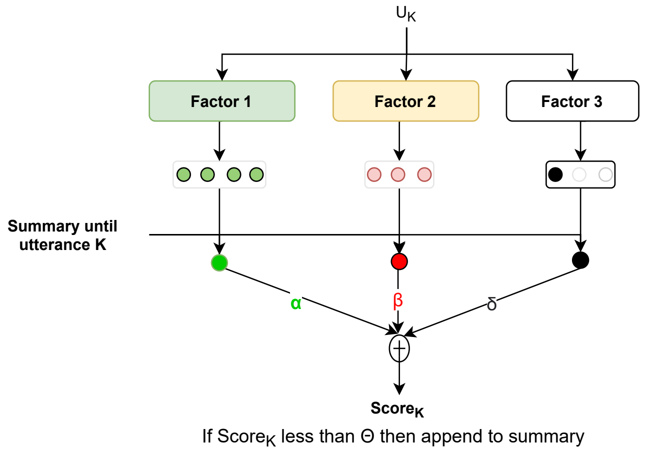
to get sentence representations. Cluster representation is then determined of each utterance using the unsupervised K-Means algorithm (Wagstaff et al., 2001). K-Means is an iterative algorithm that aims to split a data set into distinct, separate clusters with each data point in a single cluster. It ensures that the differences in utterances are identified and separated in clusters while also keeping similar utterances as close as possible. It determines which data points are assigned to which cluster by minimizing the sum squared distance of the clusters. We determined that five clusters provided optimal results and one-hot representation of five clusters were used for all of the algorithms.

6.3 Method

Most extractive summarizers generally do three tasks: they form a representation of the original text, rank the sentences accordingly, and selection of a summary comprising of a number of sentences. We explored feature-augmentation of existing widely used extractive summarizers in order to increase its efficacy in non-redundant, yet representative summarization. Specifically, we propose factored augmentations of maximum marginal relevance (MMR), Luhn and TextRank. For each summary we generate, we cap its length to 6% of the total original text length, as we find this to be enough information to provide adequate context without becoming too overwhelming in length for the reader.

MMR: Maximum Marginal Ranking

MMR produces summaries that emphasize diverse information in which similar phrases are not often repeated and a variety of information is covered in the summary (Carbonell & Goldstein, 1998). It works to decrease redundancy by doing an iterative greedy selection in which a new sentence selected for the summary is different from previously selected sentences. It measures similarity based on cosine similarity (Xie & Liu, 2008). MMR determines what utterances should be included in the final utterances list by weighing its query relevance against whether the utterance contains novel information. The determination of whether information is novel ensures that the sentences that is considered to be added to the summary is dissimilar to the previously selected sentences. By considering the similarity of the current utterance with the document and already selected key-phrases, redundancy is minimized and relevance is maximized. MMR is especially strong in preventing similar phrases that are far apart in the text from getting included in the generated



U_K
Factor 1
Factor 2
Factor 3
Summary until utterance K
α
β
δ
Score_K
If Score_K less than Θ then append to summary

Figure 6.3: Data flow diagram for MMR algorithm which can incorporate additional factors. In our work we explore use of three factors: TF-IDF, Pyscholinguistic statistics and unsupervised topic models.

summary, as similar phrases are not grouped or included in the summary (Xia et al., 2015).

Equation 6.3 shows the formula for MMR. To incorporate the LIWC values and topic modeling representation, we altered the calculation of cosine similarity by summing the original cosine similarity score along with the cosine similarity determined between the augmented embeddings. Thus, we altered the calculation of the MMR score.

$$\text{MMR} = \text{Arg. } \max_{S_i \in R/S}[\lambda Sim_{ft}(S_i, Q) - (1-\lambda) \max_{S_j \in S} Sim_{ft}(S_i, S_j))] \quad (6.1)$$

Where C is a utterance collection (or utterance stream); Q is a query or user profile; $R = IR(C, Q, \theta)$, i.e., the ranked list of utterances retrieved by an IR system, given C and Q and a relevance threshold θ, below which it will not retrieve utterances (θ can be degree of match or number of utterances); S is the subset of documents in R already selected; $R\S$ is the set difference, i.e, the set of as yet unselected documents in R; Sim_{ft} is the similarity metric which is obtained by a weighted sum of all the factors and is used in utterance retrieval and relevance ranking between utterances (passages) and a query.

$$\begin{aligned} Sim_{ft}(S_i, Q) = \alpha * Sim_{text}(S_i, Q) + \beta * Sim_{liwc}(S_i, Q) \\ + \gamma * Sim_{topic}(S_i, Q) \end{aligned} \tag{6.2}$$

In our work we give equal weights ($\alpha = 0.33, \beta = 0.33, \gamma = 0.33$) to the score of text-based similarity and LIWC based similarity, thus producing a summary balanced in diversity (due to MMR) and psycholinguistic information (LIWC score). $Sim_{text}(S_i, Q)$ is based on lexical words (the generic approach), in which we compare the current sentence (D_i) with the sentences already included in the summary (Q). Similarly, $Sim_{liwc}(S_i, Q)$ is the factor we introduce which is calculated using LIWC-similarity between the current sentence and the sentences created which are already in the summary. We add the topic-models (Cluster) in the same way, in which we introduce another similarity term based on topic models (by checking if this topic model is already represented well in the summary). The final similarity score with all three features is shown in Equation 2.

Luhn

Luhn is one of the most widlely used algorithm for text summarization (Luhn, 1958). It is designed with the assumption that the frequency of a word in text correlates with its importance. Thus, a sentence that has a lot of high-frequency words (also known as stopwords) or has a lot of words that do not often occur, does not have important meaning and should not be included in the summary. Luhn works in two stages: determining the significance of words in relation to the meaning of the document via frequency analysis and then determining common words that are not among the most common and are still important Fein et al. (1999).

Luhn then assigns a sentence or an utterance score $Luhn_i$ based on important lexical words which appear in that sentence. For example: If a sentence has N words and K (subset of N) of them are among the list of relevant words then $Luhn_{text(i)}$ of a sentence i is given by squared ratio of frequency of important K words dived by the total number of words in that sentence. In our work we provide two additional sentence level scores bease on two factors: $Luhn_{liwc(i)}$, where marks ratio of squared frequency of words belonging any of categories listed in Table 1 and total number of words in a sentence. Similarly, $Luhn_{topic(i)}$ is the ratio of squared number of topics words in a sentence belong to divided by total number of topics in a document. Final score for a each sentence i is then computed by weighted mean of scores from each factor using hyperparamters set by user for each factor

$$Luhn_i = Luhn_{text(i)} + Luhn_{liwc(i)} + Luhn_{topic(i)} \tag{6.3}$$

Luhn algorithm returns final summary of a document or a conversation by sorting the sentences according $Luhn_i$ score, where similar to previous algorithm we select 6% of total sentences to form a summary. It should be noted that summary still has sentences in the chronological order as we believe it's easier to understand a summary if it follows the chronology of a conversation.

Text Rank

TextRank is a general purpose, graph based ranking algorithm for natural language processing. Graph-based ranking algorithms are a way for deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. TextRank is very well-suited for applications involving entire sentences, since it allows for a ranking over text units that is recursively computed based on information drawn from the entire text Mihalcea & Tarau (2004). To apply TextRank, we first build a graph associated with the text, in which the graph vertices are representative for the units to be ranked. In our case, the vertices are sentences, in which the goal is to rank entire sentences; therefore, a vertex is added to the graph for each sentence in the text. TextRank uses scores provided by PageRankHaveliwala (1999) to provide a score for each vertex (each sentence) and then searches for the minimum spanning tree which has the highest cost. Thus, it collects important sentences of the document and generates a summary.

In order to exploit additional features, we provide scores related to LIWC and topic models in addition to the score provided by PageRank in a similar fashion as described in previous subsection.

6.4 Evaluation

To test our augmented summarizers with incorporates multiple factors we use a standardized conversational data set. We first test summarizer performance automatically with ROUGE score in relation to the human-produced gold summaries to attain quantitative results. We then test the quality of the summarizer manually with a human trial, attaining qualitative results.

Gold Summary	Summary w/ LIWC + Clusters
today is our third meeting. It will be about the conceptual design. We decided not to go for speech recognition technologies because of some reasons. we are not decided about the use of LCD screen on the remote control because of costs.	which fruit are you thinking of? I haven't thought of any particular fruit, but the general aspect of the remote control may could remind some kind of vegetable, some kind of instead of vegetable, some natural object or something. maybe you can display a banana on the LCD. I'm not, I'm not really sure if that would really appeal to everyone though, maybe just to fashion gurus, like maybe just like a little bit a little fruit picture somewhere in the corner, but I don't know about I don't know how ergonomic a, an orange is.

Table 6.2: First four sentences of summary about a meeting that discussed the potential fruit logo and design of a new remote

6.4.1 Data

We evaluate our proposed feature-augmented extractive summarization on a generic meeting corpus, AMI meeting corpus. The AMI meeting corpus(Carletta et al., 2005) is a collection of 139 meeting records where groups of people are engaged in a 'roleplay' as a team and each speaker assumes a certain role in a team (e.g. project manager). The meetings are on the topic of developing meeting browsing technology. The meetings are composed of both real interactions and acted-out scenarios of a design team creating a one-day design project. We use 80 transcripts for evaluation and remaining 60 transcripts for tuning and development purposes. This data is well-suited for this task because each transcript is a multi-person conversation, allowing us to explore factored extractive summarization in a novel setting that is more likely to have psycholinguistic and topic-based dictation.

6.4.2 Automatic Evaluation

We report ROUGE (Lin & Hovy, 2002) scores between the gold and generated summary for each augmentation of our summarizers in order to determine whether the influence of LIWC data and topic modeling created summaries closer to the human-generated versions than the unaugmented algorithms'. We include F1-measure, precision and recall scores of ROUGE-1 and ROUGE-2. ROUGE-1 refers

Algorithms	Features	F1 Rouge 1	Recall 1	Precision 1	F1 Rouge 2	Recall 2	Precision 2
MMR	N/A	0.21	0.14	0.75	0.09	0.06	0.37
MMR	LIWC	0.23	0.15	0.79	0.12	0.07	0.46
MMR	LIWC + Clusters	0.23	0.15	0.80	0.12	0.07	0.48
Luhn	N/A	0.45	0.36	**0.82**	0.29	0.22	0.56
Luhn	LIWC	**0.47**	**0.38**	0.80	**0.30**	**0.23**	0.54
Luhn	LIWC + Clusters	0.39	0.29	**0.82**	0.23	0.16	0.52
TextRank	N/A	0.36	0.30	0.76	0.22	0.15	0.52
TextRank	LIWC	0.40	0.35	0.80	0.25	0.20	**0.58**
TextRank	LIWC + Clusters	0.40	0.35	0.81	0.26	0.20	**0.58**

Table 6.3: Results for Automatic Evaluation: Comparing each extractive algorithm with it's feature-augmented variant

to unigram overlap between the generated and gold summaries, ROUGE-2 refers to bigram overlap.Liu & Liu (2010).

Initially, we ran the unchanged summarizers of MMR, Luhn and TextRank in order to establish baseline performance. We then incorporated psycho-linguistic information (based on LIWC) to influence the summary to be more representative of sentiment-heavy statements. Finally, we incorporated unsupervised topic clusters to ensure a diversity of content was represented and to reduce redundancy.

The results of these comparisons are shown in Table 6.3. Results show that feature-augmented scores can generate summaries closer to the gold summaries. ROUGE scores slightly improved for all three algorithms compared, with the exception of Luhn when it was incorporated with both LIWC and topic modeling.

6.4.3 Manual Evaluation

10 transcripts were randomly selected and summarized in four different ways: *Gold summary, Baseline Algorithm, Augmented with psycho-linguistic information* and *Augmented with psycho-linguistic and topic modeling information.* Five of the transcripts were assessed and augmented with MMR as the baseline algorithm and Luhn was assessed for the other five. 28 human evaluators assessed the efficacy

Algorithm	Feature	Gold	Original	w/L	w/L+C
MMR	1	2%	23%	42%	26%
Luhn	1	2%	23%	49%	20%
MMR	2	7%	12%	32%	42%
Luhn	2	9%	15%	29%	40%
MMR	3	5%	17%	22%	49%
Luhn	3	5%	11%	27%	50%
MMR	4	6%	11%	14%	63%
Luhn	4	7%	7%	13%	67%

Table 6.4: % User Preference of Summaries Along Dimensions, 1 = Lack of Redundancy, 2 = Context, 3 = Content, 4 = Sentiment. Gold refers to human created summaries. L = incorporation of LIWC features, L + C = incorporation of LIWC and topic modeling information

of the MMR series and the Luhn series. In order to understand the context of the summaries, these individuals were educated upon the purpose of each meeting prior to evaluation. To ensure proper appraisal of these summaries, evaluators were asked to summarize the main content of the summary.

Each participant chose the summary of each transcript which fit best for each of the following four categories:

- Least redundancy
- Best context
- Best content
- Highest emotional information

The average results are shown in Table 6.4. Results show that summaries became more preferable with each augmentation to each algorithm, increasing in all four categories drastically. Even more notable is that the gold, human-produced summary was not determined to be the best in any category.

6.5 Discussion

For this context of conversational data, ROUGE scores are not always reliable for quality of summary because they depend on the human producing the gold summaries prioritizing the same type of content as the summarizer. Additionally, in conversational settings in particular, it is important for summaries to be succinct for future reference. The AMI gold summaries were extremely lengthy. ROUGE scores work in a fashion such that if a summary is more concise than the human-generated summary, the ROUGE score will still mark the summary lower as it is dissimilar. Quality in this situation is much better assessed by human evaluation.

Our summaries that incorporated psycho-linguistics and topic-modeling outperformed the human-produced gold summaries and the baseline summaries in user trials in all four categories. Although the ROUGE score did not greatly improve with each augmentation, the summaries we generated with these augmentations were preferable to humans. In fact, user results indicate that they are superior to gold summaries. Because our generated summaries did not get much closer in similarity to the human-generated summaries, the ROUGE score did not change significantly; however, the quality did improve, as confirmed by the human evaluations. The summaries that included the LIWC and cluster information was most preferable for every feature besides the metric measuring which summary had the least redundancy. In this case, the algorithm that solely incorporated LIWC information was preferable. This shows that ROUGE score might not always be a reliable metric for determining summary quality. Our augmentations eliminate the noise that is typical in even human-generated summaries, emphasizing the most sentiment-heavy portions while also providing a diversity in content, as shown by the large jump in preference for all four categories users rated for the augmented algorithms, with the gold summaries scoring low in every test. By concentrating psycho-linguistic/sentiment-heavy, diverse content into a concise summary, we are able to produce a more preferable summaries than the gold summaries.

6.6 Conclusions and Future Work

In this work, we presented results towards how additional features can be incorporated in order to extract and summarize crucial information from conversational transcripts. We applied the original extractive algorithms to AMI data, and then increasingly altered versions of the algorithms in order to determine which alterations can enhance the redundancy and diversity of its summarization. In addition, we determined the similarity of our generated summaries with gold summaries and contrasted this with user feedback. From this, we determined that the summaries that integrated psycho-linguistic variables and topic modeling are qualitatively stronger than the gold and baseline summaries. It is an interesting conclusion that highly emotive or sentiment-heavy sentence correlates with more important information for a summary. In fact, this incorporation of sentiment data and topic modeling could potentially be translated to other domains that value understanding emotional state and prioritize diversity of information, such as counseling. It would be interesting to further explore these conclusions on more types of data sets and to incorporate different, potentially beneficial features in summarization. We plan to explore these problem areas in our future efforts.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation, we proposed and experimentally investigated several machine learning paradigms for the leveraging auxiliary information either using multi-modal or language modeling based approaches. We show using inherent semantic linguistic structure (understanding words to understand sentences, sentences for documents) makes it easier for machine to learn using limited amount of supervision. In chapter 2, we show that acoustic-prosodic cues can aid lexical text for better automatic understanding and prediction of behaviors of psychotherapy. Our proposed methods show that variation in acoustic-prosodic information (pitch, pause, speech-rate, jitter) between words provides discriminatory information for behavior code prediction. In chapter 3 we propose a novel method which doesn't use any transcription but only word segmentation to learn representation for a spoken word. We also show that self-supervised representations learning by regenerating context speech words can be used for transcription-free prediction of behavior codes. We believe this transcription-free approach can be extended to multiple SLU domains in the future. Thus walking towards a near future where Spoken Language Understanding (SLU) requires limited human supervision and exploits both semantic and acoustic-prosodic information.

In Chapter 4 and 5 we show that joint multitask training which exploits additional monolingual data can learn task specific encoder using limited amount of supervised corpus. We propose a novel architecture which learns multilingual sentence representations along with a multilingual word representation objective. Our proposed method can transparently exchange information between the two tasks. Ablation studies suggest this architecture can exploit 3-4 times additional monolingual data than the costly parallel corpus to learn high-quality limited resource sentence representations. We also show this similar architecture which learns a document classifier along with a language modeling objective can help with improved limited resource document classification. Lastly, we propose a factored-extractive conversation summarization methods where we show that psycholinguistics and unsupervised topic models can help with better contextualized and affect aware text summarization.

7.2 Future Directions

7.2.1 Multitask End-2-End Spoken Utterance Labeling

This idea is motivated from efforts from Speech2VecChung & Glass (2018); Baevski et al. (2020) to learn generic speech and music encoder. Most of the self-supervised methods generally use a self-learning objective e.g: masking of neighbouring time-series samples and then either generating or predicting context. However, they assume to be trained on clean mono-channel speech. There has been limited or no-work on learning multi-purpose universal speech signal representations.

Figure 7.1 proposes such a model which can take into multi-channel data, differentiate noise from important information. The system will be fine tuned by variety of information from multiple aspects of speech signal (ranging from tasks that can cover linguistic information along with acoustic-prosodic information).

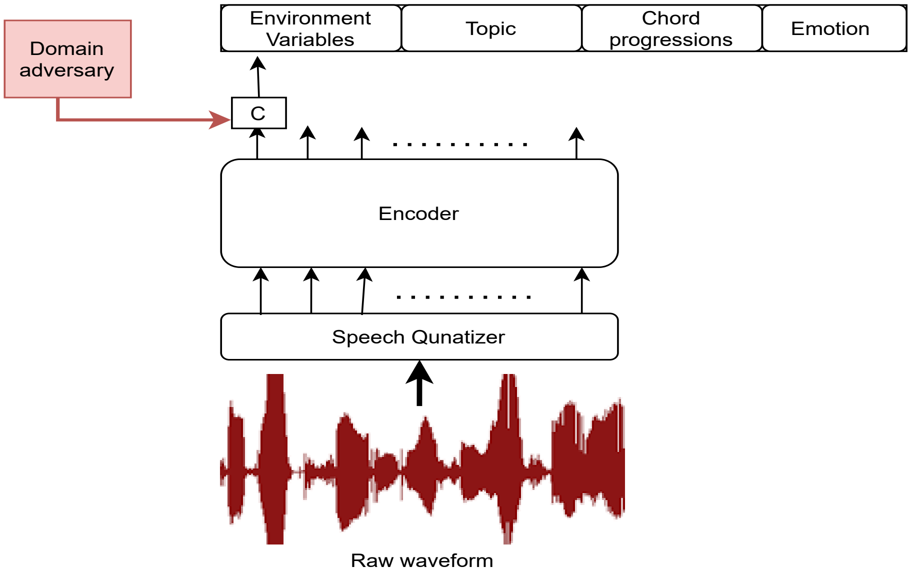


Figure 7.1: Here a speech segment is first quantized (discrete or continues) either using an ML technique or functional over a fixed window size. Then quantized speech is passed through encoder (commonly used: transformer). The whole system is optimized using sparse multitask training of self-supervised and supervised approaches.

Described below is the loss objective. Here is B is a fixed length hyper-parameter matrix to provide weights for N tasks, which can also be learnt during training.

$$Loss(s) = SelfSupervisionLoss + [B] * \sum_{i=1}^{N} Task_i \tag{7.1}$$

7.2.2 Universal End-2-End Spoken utterance representations

There are many recent methods for learning unsupervised or self-supervised spoken language representations. However when it comes to making scalable systems research still prefers making an ASR. There is limited success in making an multilingual ASR which can transcribe multiple languages. In reality, when it comes to SLU ASR generally helps in getting standardized text representations. This text is then represented in vector space for further processing. However when it comes to making scalable systems research still prefers making an ASR. There are primary two reasons for this practice, firstly because research has matured in making general purpose ASR which can transcribe speech into text. Secondly, traditionally speech signal processing has followed HMM-GMM kind of approaches to making ASR. Recent works like —– has worked towards conntecting linguistics theory used in natural language processing to speech signal processing. Thus working towards an End-2-End learning of semantics. This opens doors to conduct research on developing methods for multilingual general unsupervised representations of speech which can then be used for SLU from multiple facets.

Reference List

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al. (2016) Tensorflow: A system for large-scale machine learning In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283.

Alguliev R, Aliguliyev R et al. (2009) Evolutionary algorithm for extractive text summarization. *Intelligent Information Management* 1:128.

Ammar W, Mulcaire G, Ballesteros M, Dyer C, Smith NA (2016) Many languages, one parser. *arXiv preprint arXiv:1602.01595* .

Amodei D, Ananthanarayanan S, Anubhai R, Bai J, Battenberg E, Case C, Casper J, Catanzaro B, Cheng Q, Chen G et al. (2016) Deep speech 2: End-to-end speech recognition in english and mandarin In *International conference on machine learning*, pp. 173–182.

Arik SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J et al. (2017) Deep voice: Real-time neural text-to-speech In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 195–204. JMLR. org.

Artetxe M, Schwenk H (2018) Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464* .

Atkins DC, Steyvers M, Imel ZE, Smyth P (2014) Scaling up the evaluation of psychotherapy: evaluating motivational interviewing fidelity via statistical text classification. *Implementation Science* 9:49.

Baer JS, Wells EA, Rosengren DB, Hartzler B, Beadnell B, Dunn C (2009) Agency context and tailored training in technology transfer: A pilot evaluation of motivational interviewing training for community counselors. *Journal of Substance Abuse Treatment* 37:191–202.

Baevski A, Zhou H, Mohamed A, Auli M (2020) wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477* .

Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Barone AVM (2016) Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. *arXiv preprint arXiv:1608.02996* .

Baxter J (1997) A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning* 28:7–39.

Bengio Y, Corrado G (2015) Bilbowa: Fast bilingual distributed representations without word alignments .

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

Boersma P (2006) Praat: doing phonetics by computer. *http://www. praat. org/* .

Can D, Atkins DC, Narayanan SS (2015) A dialog act tagging approach to behavioral coding: A case study of addiction counseling conversations In *Sixteenth Annual Conference of the International Speech Communication Association*.

Can D, Georgiou PG, Atkins DC, Narayanan SS (2012) A case study: Detecting counselor reflections in psychotherapy for addictions using linguistic features In *Thirteenth Annual Conference of the International Speech Communication Association*.

Cao J, Tanana M, Imel ZE, Poitras E, Atkins DC, Srikumar V (2019) Observing dialogue in therapy: Categorizing and forecasting behavioral codes. *arXiv preprint arXiv:1907.00326* .

Carbonell J, Goldstein J (1998) The use of mmr, diversity-based reranking for reordering documents and producing summaries In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336.

Carletta J, Ashby S, Bourban S, Flynn M, Guillemot M, Hain T, Kadlec J, Karaiskos V, Kraaij W, Kronenthal M et al. (2005) The ami meeting corpus: A pre-announcement In *International workshop on machine learning for multimodal interaction*, pp. 28–39. Springer.

Chan W, Jaitly N, Le Q, Vinyals O (2016) Listen, attend and spell: A neural network for large vocabulary conversational speech recognition In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964. IEEE.

Chandar S, Lauly S, Larochelle H, Khapra M, Ravindran B, Raykar VC, Saha A (2014) An autoencoder approach to learning bilingual word representations In *Advances in Neural Information Processing Systems*, pp. 1853–1861.

Chen D, Manning CD (2014) A fast and accurate dependency parser using neural networks. In *EMNLP*, pp. 740–750.

Chen M (2017) Efficient vector representation for documents through corruption. *arXiv preprint arXiv:1707.02377* .

Chen Z, Singla K, Gibson J, Can D, Imel ZE, Atkins DC, Georgiou P, Narayanan S (2019) Improving the prediction of therapist behaviors in addiction counseling by exploiting class confusions In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6605–6609. IEEE.

Cheung L, Gowda T, Hermjakob U, Liu N, May J, Mayn A, Pourdamghani N, Pust M, Knight K, Malandrakis N et al. (2017) Elisa system description for lorehlt 2017. *Proc. Low Resource Human Lang. Technol* pp. 51–59.

Chidambaram M, Yang Y, Cer D, Yuan S, Sung YH, Strope B, Kurzweil R (2018) Learning cross-lingual sentence representations via a multi-task dual-encoder model. *arXiv preprint arXiv:1810.12836* .

Chorowski JK, Bahdanau D, Serdyuk D, Cho K, Bengio Y (2015) Attention-based models for speech recognition In *Advances in neural information processing systems*, pp. 577–585.

Christensen A, Atkins DC, Berns S, Wheeler J, Baucom DH, Simpson LE (2004) Traditional versus integrative behavioral couple therapy for significantly and chronically distressed married couples. *Journal of consulting and clinical psychology* 72:176.

Christianson C, Duncan J, Onyshkevych B (2018) Overview of the darpa lorelei program. *Machine Translation* 32:3–9.

Chung YA, Glass J (2018) Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *arXiv preprint arXiv:1803.08976* .

Chung YA, Weng WH, Tong S, Glass J (2019) Towards unsupervised speech-to-text translation In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7170–7174. IEEE.

Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM.

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A (2017) Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364* .

Coulmance J, Marty JM, Wenzek G, Benhalloum A (2016) Trans-gram, fast cross-lingual word-embeddings. *arXiv preprint arXiv:1601.02502* .

Creed TA, Frankel SA, German RE, Green KL, Jager-Hyman S, Taylor KP, Adler AD, Wolk CB, Stirman SW, Waltman SH et al. (2016) Implementation of transdiagnostic cognitive therapy in community behavioral health: The beck community initiative. *Journal of consulting and clinical psychology* 84:1116.

De Mori R, Bechet F, Hakkani-Tur D, McTear M, Riccardi G, Tur G (2008) Spoken language understanding. *IEEE Signal Processing Magazine* 25:50–58.

Desot T, Portet F, Vacher M (2019) Slu for voice command in smart home: comparison of pipeline and end-to-end approaches In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 822–829. IEEE.

Duong L, Kanayama H, Ma T, Bird S, Cohn T (2016) Learning crosslingual word embeddings without bilingual corpora. *arXiv preprint arXiv:1606.09403* .

Faruqui M, Dyer C (2014) Improving vector space word representations using multilingual correlation Association for Computational Linguistics.

Fein RA, Dolan WB, Messerly J, Fries EJ, Thorpe CA, Cokus SJ (1999) Document summarizer for word processors US Patent 5,924,108.

Ganesh P, Dingliwal S (2019) Abstractive summarization of spoken and written conversation. *arXiv preprint arXiv:1902.01615* .

Gerani S, Mehdad Y, Carenini G, Ng R, Nejat B (2014) Abstractive summarization of product reviews using discourse structure In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1602–1613.

Giannakopoulos T (2015) pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one* 10:e0144610.

Gibson J, Atkins D, Creed T, Imel Z, Georgiou P, Narayanan S (2019) Multi-label multi-task deep learning for behavioral coding. *IEEE Transactions on Affective Computing* .

Gibson J, Can D, Georgiou P, Atkins D, Narayanan S (2017a) Attention networks for modeling behavior in addiction counseling In *In Proceedings of Interspeech.*

Gibson J, Can D, Georgiou PG, Atkins DC, Narayanan SS (2017b) Attention networks for modeling behaviors in addiction counseling. In *INTERSPEECH*, pp. 3251–3255.

Glynn LH, Moyers TB (2012) Manual for the client language easy rating (clear) coding system: Formerly 'motivational interviewing skill code (misc) 1.1'. *Retrieved November* 13:2017.

Gouws S, Bengio Y, Corrado G (2015) Bilbowa: Fast bilingual distributed representations without word alignments In *International Conference on Machine Learning*, pp. 748–756.

Guo Y, Xiao M (2012) Cross language text classification via subspace co-regularized multi-view learning. *arXiv preprint arXiv:1206.6481* .

Gupta O, Raviv D, Raskar R (2017) Multi-velocity neural networks for facial expression recognition in videos. *IEEE Transactions on Affective Computing* .

Haghani P, Narayanan A, Bacchiani M, Chuang G, Gaur N, Moreno P, Prabhavalkar R, Qu Z, Waters A (2018) From audio to semantics: Approaches to end-to-end spoken language understanding In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 720–726. IEEE.

Haque A, Guo M, Verma P, Fei-Fei L (2019) Audio-linguistic embeddings for spoken sentences In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7355–7359. IEEE.

Hardy GE, Llewelyn S (2015) Introduction to psychotherapy process research In *Psychotherapy research*, pp. 183–194. Springer.

Hashimoto K, Xiong C, Tsuruoka Y, Socher R (2016) A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587* .

Haveliwala T (1999) Efficient computation of pagerank Technical report, Stanford.

Hermann KM, Blunsom P (2013) Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173* .

Hermann KM, Blunsom P (2014) Multilingual models for compositional distributed semantics. *arXiv preprint arXiv:1404.4641* .

Heyman RE, Lorber MF, Eddy JM, West TV (2014) Behavioral observation and coding. .

Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J (2001) Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural computation* 9:1735–1780.

Hong K, Nenkova A (2014) Improving the estimation of word importance for news multi-document summarization In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 712–721.

Iwano K, Hirose K (1999) Prosodic word boundary detection using statistical modeling of moraic fundamental frequency contours and its use for continuous speech recognition In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, Vol. 1, pp. 133–136. IEEE.

Jabaian B, Besacier L, Lefèvre F (2010) Investigating multiple approaches for slu portability to a new language In *Eleventh Annual Conference of the International Speech Communication Association*.

Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, Thorat N, Viégas F, Wattenberg M, Corrado G et al. (2016) Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558* .

Junqua JC, Mak B, Reaves B (1994) A robust algorithm for word boundary detection in the presence of noise. *IEEE Transactions on speech and audio processing* 2:406–412.

Kidd E, Tennant E, Nitschke S (2015) Shared abstract representation of linguistic structure in bilingual sentence comprehension. *Psychonomic Bulletin & Review* 22:1062–1067.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kiros R, Zhu Y, Salakhutdinov RR, Zemel R, Urtasun R, Torralba A, Fidler S (2015) Skip-thought vectors In *Advances in neural information processing systems*, pp. 3294–3302.

Klementiev A, Titov I, Bhattarai B (2012) Inducing crosslingual distributed representations of words .

Koehn P (2005) Europarl: A parallel corpus for statistical machine translation In *MT summit*, Vol. 5, pp. 79–86.

Koehn P (2009) *Statistical machine translation* Cambridge University Press.

Lauly S, Boulanger A, Larochelle H (2014) Learning multilingual word representations using a bag-of-words autoencoder. *arXiv preprint arXiv:1401.1803* .

Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In *ICML*, Vol. 14, pp. 1188–1196.

Lee KF (1988) *Automatic speech recognition: the development of the SPHINX system*, Vol. 62 Springer Science & Business Media.

Lilleberg J, Zhu Y, Zhang Y (2015) Support vector machines and word2vec for text classification with semantic features In *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pp. 136–140.

Lin CY, Hovy E (2002) Manual and automatic evaluation of summaries In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pp. 45–51. Association for Computational Linguistics.

Liu F, Liu Y (2010) Exploring correlation between rouge and human evaluation on meeting summaries. *IEEE Transactions on Audio, Speech, and Language Processing* 18:187–196.

Liu F, Flanigan J, Thomson S, Sadeh N, Smith NA (2018) Toward abstractive summarization using semantic representations. *arXiv preprint arXiv:1805.10399* .

Liu P, Qiu X, Huang X (2016) Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101* .

Liu X, Gao J, He X, Deng L, Duh K, Wang YY (2015) Representation learning using multi-task deep neural networks for semantic classification and information retrieval .

Lugosch L, Ravanelli M, Ignoto P, Tomar VS, Bengio Y (2019) Speech model pre-training for end-to-end spoken language understanding. *arXiv preprint arXiv:1904.03670* .

Luhn HP (1958) The automatic creation of literature abstracts. *IBM Journal of research and development* 2:159–165.

Luong T, Pham H, Manning CD (2015) Bilingual word representations with monolingual quality in mind In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 151–159.

Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 142–150. Association for Computational Linguistics.

Malandrakis N, Glembek O, Narayanan SS (2017) Extracting situation frames from non-english speech: Evaluation framework and pilot results. In *INTERSPEECH*, pp. 2123–2127.

Malandrakis N, Ramakrishna A, Martinez V, Sorensen T, Can D, Narayanan S (2018) The elisa situation frame extraction for low resource languages pipeline for lorehlt'2016. *Machine Translation* 32:127–142.

Marechal C, Mikolajewski D, Tyburek K, Prokopowicz P, Bougueroua L, Ancourt C, Wegrzyn-Wolska K (2019) Survey on ai-based multimodal methods for emotion detection.

Margolin G, Oliver PH, Gordis EB, O'hearn HG, Medina AM, Ghosh CM, Morland L (1998) The nuts and bolts of behavioral observation of marital and family interaction. *Clinical child and family psychology review* 1:195–213.

Mihalcea R (2004) Graph-based ranking algorithms for sentence extraction, applied to text summarization In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 170–173.

Mihalcea R, Banea C, Wiebe J (2007) Learning multilingual subjective language via cross-lingual projections In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 976–983.

Mihalcea R, Tarau P (2004) Textrank: Bringing order into text In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411.

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality In *Advances in neural information processing systems*, pp. 3111–3119.

Miller WR, Moyers TB, Ernst D, Amrhein P (2003) Manual for the motivational interviewing skill code (misc). *Unpublished manuscript. Albuquerque: Center on Alcoholism, Substance Abuse and Addictions, University of New Mexico* .

Miller WR, Rose GS (2009) Toward a theory of motivational interviewing. *American psychologist* 64:527.

Mogadala A, Rettinger A (2016) Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification In *Proceedings of NAACL-HLT*, pp. 692–702.

Mroueh Y, Marcheret E, Goel V (2015) Deep multimodal learning for audio-visual speech recognition In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 2130–2134. IEEE.

Muis AO, Otani N, Vyas N, Xu R, Yang Y, Mitamura T, Hovy E (2018) Low-resource cross-lingual event type detection via distant supervision with minimal effort In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 70–82.

Narayanan S, Georgiou PG (2013) Behavioral signal processing: Deriving human behavioral informatics from speech and language. *Proceedings of the IEEE* 101:1203–1233.

Newmark P (1998) *More paragraphs on translation* Multilingual matters.

Ochshorn R, Hawkins M (2016) Gentle: A forced aligner.

Oord Avd, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* .

Oya T, Mehdad Y, Carenini G, Ng R (2014) A template-based abstractive meeting summarization: Leveraging summary and source text relationships In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pp. 45–53.

Pan JZ, Vetere G, Gomez-Perez JM, Wu H (2017) *Exploiting linked data and knowledge graphs in large organisations* Springer.

Panayotov V, Chen G, Povey D, Khudanpur S (2015) Librispeech: an asr corpus based on public domain audio books In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. IEEE.

Papadopoulos P, Travadi R, Vaz C, Malandrakis N, Hermjakob U, Pourdamghani N, Pust M, Zhang B, Pan X, Lu D et al. (2017) Team elisa system for darpa lorelei speech evaluation 2016. In *INTERSPEECH*, pp. 2053–2057.

Paszke A, Gross S, Chintala S, Chanan G (2017) Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration* 6.

Paulus R, Xiong C, Socher R (2017) A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* .

Pennebaker JW, Francis ME, Booth RJ (2001) Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates* 71:2001.

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In *EMNLP*, Vol. 14, pp. 1532–1543.

Pérez-Rosas V, Mihalcea R, Resnicow K, Singh S, An L, Goggin KJ, Catley D (2017) Predicting counselor behaviors in motivational interviewing encounters In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 1128–1137.

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* .

Pham H, Luong MT, Manning CD (2015) Learning distributed representations for multilingual text sequences In *Proceedings of NAACL-HLT*, pp. 88–94.

Pham NT, Kruszewski G, Lazaridou A, Baroni M (2015) Jointly optimizing word representations for lexical and sentential tasks with the c-phrase model. In *ACL (1)*, pp. 971–981.

Pittaras N, Karkaletsis V (2019) A study of semantic augmentation of word embeddings for extractive summarization In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, pp. 63–72, Varna, Bulgaria. INCOMA Ltd.

Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, Hannemann M, Motlicek P, Qian Y, Schwarz P et al. (2011) The kaldi speech recognition toolkit In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society.

Prasojo RE, Kacimi M, Nutt W (2018) Modeling and summarizing news events using semantic triples In *European Semantic Web Conference*, pp. 512–527. Springer.

Pruette JR (2013) Autism diagnostic observation schedule-2 (ados-2). *Google Scholar* .

Qian Y, Ubale R, Ramanaryanan V, Lange P, Suendermann-Oeft D, Evanini K, Tsuprun E (2017) Exploring asr-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 569–576. IEEE.

Ramesh R, Rajan B (2019) Extractive text summarization using graph based ranking algorithm and mean shift clustering. *Available at SSRN 3439357* .

Ravuri S, Stolcke A (2015) Recurrent neural network and lstm models for lexical utterance classification In *Sixteenth Annual Conference of the International Speech Communication Association*.

Ruder S (2017a) An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* .

Ruder S (2017b) A survey of cross-lingual embedding models. *CoRR* abs/1706.04902.

Schneider S, Baevski A, Collobert R, Auli M (2019) wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862* .

Schuller B, Lang M, Rigoll G (2002) Multimodal emotion recognition in audiovisual communication In *Proceedings. IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 745–748. IEEE.

Schwenk H, Tran K, Firat O, Douze M (2017) Learning joint multilingual sentence representations with neural machine translation. *arXiv preprint arXiv:1704.04154* .

Sebe N, Cohen I, Huang TS (2005) Multimodal emotion recognition In *Handbook of Pattern Recognition and Computer Vision*, pp. 387–409. World Scientific.

Serdyuk D, Wang Y, Fuegen C, Kumar A, Liu B, Bengio Y (2018) Towards end-to-end spoken language understanding In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5754–5758. IEEE.

Servan C, Camelin N, Raymond C, Béchet F, De Mori R (2010) On the use of machine translation for spoken language understanding portability In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5330–5333. IEEE.

Sharma B, Madhavi M, Li H (2021) Leveraging acoustic and linguistic embeddings from pretrained speech and language models for intent classification In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7498–7502. IEEE.

Shi L, Mihalcea R, Tian M (2010) Cross language text classification by model translation and semi-supervised learning In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1057–1067. Association for Computational Linguistics.

Singla K, Can D, Narayanan S (2018) A multi-task approach to learning multilingual representations In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 214–220.

Singla K, Chen Z, Atkins D, Narayanan S (2020) Towards end-2-end learning for predicting behavior codes from spoken utterances in psychotherapy conversations In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3797–3803.

Singla K, Chen Z, Flemotomos N, Gibson J, Can D, Atkins DC, Narayanan S (2018) Using prosodic and lexical information for learning utterance-level behaviors in psychotherapy. In *Interspeech*, pp. 3413–3417.

Singla K, Narayanan S (2020) Multitask learning for darpa lorelei's situation frame extraction task In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8149–8153. IEEE.

Siriwardhana S, Reis A, Weerasekera R, Nanayakkara S (2020) Jointly fine-tuning" bert-like" self supervised models to improve multimodal speech emotion recognition. *arXiv preprint arXiv:2008.06682* .

Socher R, Manning CD, Ng AY (2010) Learning continuous phrase representations and syntactic parsing with recursive neural networks In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pp. 1–9.

Søgaard A, Agić Ž, Alonso HM, Plank B, Bohnet B, Johannsen A (2015) Inverted indexing for cross-lingual nlp In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015).*

Søgaard A, Goldberg Y (2016) Deep multi-task learning with low level tasks supervised at lower layers In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2, pp. 231–235.

Soltau H, Liao H, Sak H (2016) Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition. *arXiv preprint arXiv:1610.09975* .

Subramanian S, Trischler A, Bengio Y, Pal CJ (2018) Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079* .

Tafforeau J, Bechet F, Artières T, Favre B (2016) Joint syntactic and semantic analysis with a multitask deep learning framework for spoken language understanding. In *Interspeech*, pp. 3260–3264.

Tanana M, Hallgren KA, Imel ZE, Atkins DC, Srikumar V (2016) A comparison of natural language processing methods for automated coding of motivational interviewing. *Journal of substance abuse treatment* 65:43–50.

Tran T, Toshniwal S, Bansal M, Gimpel K, Livescu K, Ostendorf M (2017) Joint modeling of text and acoustic-prosodic cues for neural parsing. *arXiv preprint arXiv:1704.07287* .

Tsiartas A, Ghosh PK, Georgiou P, Narayanan S (2009) Robust word boundary detection in spontaneous speech using acoustic and lexical cues In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4785–4788. IEEE.

Tur G, De Mori R (2011) *Spoken language understanding: Systems for extracting semantic information from speech* John Wiley & Sons.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need In *Advances in neural information processing systems*, pp. 5998–6008.

Vulic I, Moens MF (2015) Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pp. 719–725. ACL.

Vulić I, Moens MF (2016) Bilingual distributed word representations from document-aligned comparable data. *Journal of Artificial Intelligence Research* 55:953–994.

Vyas Y, Carpuat M (2016) Sparse bilingual word representations for cross-lingual lexical entailment In *Proceedings of NAACL-HLT*, pp. 1187–1197.

Wagstaff K, Cardie C, Rogers S, Schrödl S et al. (2001) Constrained k-means clustering with background knowledge In *Icml*, Vol. 1, pp. 577–584.

Wiesner M, Liu C, Ondel L, Harman C, Manohar V, Trmal J, Huang Z, Dehak N, Khudanpur S (2018) Automatic speech recognition and topic identification for almost-zero-resource languages In *Proc. Interspeech.*

Wu L, Fisch A, Chopra S, Adams K, Bordes A, Weston J (2017a) Starspace: Embed all the things! *arXiv preprint arXiv:1709.03856* .

Wu S, Zhang D, Yang N, Li M, Zhou M (2017b) Sequence-to-dependency neural machine translation In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, pp. 698–707.

Xia L, Xu J, Lan Y, Guo J, Cheng X (2015) Learning maximal marginal relevance model via directly optimizing diversity evaluation measures In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, p. 113–122, New York, NY, USA. Association for Computing Machinery.

Xiao B, Bone D, Van Segbroeck M, Imel ZE, Atkins D, Georgiou P, Narayanan S (2014) Modeling therapist empathy through prosody in drug addiction counseling In *Proceedings of Interspeech*.

Xiao B, Can D, Gibson J, Imel ZE, Atkins DC, Georgiou PG, Narayanan SS (2016a) Behavioral coding of therapist language in addiction counseling using recurrent neural networks. In *Interspeech*, pp. 908–912.

Xiao B, Gibson J, Can D, Imel ZE, Atkins DC, Georgiou P, Narayanan SS (2016b) Behavioral coding of therapist language in addiction counseling using recurrent neural networks In *Proceedings of Interspeech*.

Xiao B, Huang CW, Imel ZE, Atkins DC, Georgiou P, Narayanan SS (2016c) A technology prototype system for rating therapist empathy from audio recordings in addiction counseling. *PeerJ Computer Science* 2.

Xiao B, Huang C, Imel ZE, Atkins DC, Georgiou P, Narayanan SS (2016d) A technology prototype system for rating therapist empathy from audio recordings in addiction counseling. *PeerJ Computer Science* 2:e59.

Xiao B, Imel ZE, Atkins D, Georgiou P, Narayanan SS (2015a) Analyzing speech rate entrainment and its relation to therapist empathy in drug addiction counseling In *Proceedings of Interspeech*.

Xiao B, Imel ZE, Georgiou PG, Atkins DC, Narayanan SS (2015b) " rate my therapist": Automated detection of empathy in drug and alcohol counseling via speech and language processing. *PloS one* 10:e0143055.

Xiao M, Guo Y (2014) Distributed word representation learning for cross-lingual dependency parsing. In *CoNLL*, pp. 119–129.

Xie S, Liu Y (2008) Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4985–4988. IEEE.

Xing C, Wang D, Liu C, Lin Y (2015) Normalized word embedding and orthogonal transform for bilingual word translation. In *HLT-NAACL*, pp. 1006–1011.

Xu P, Sarikaya R (2014) Contextual domain classification in spoken language understanding systems using recurrent neural network In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 136–140. IEEE.

Xu R, Yang Y, Liu H, Hsi A (2016) Cross-lingual text classification via model translation with limited dictionaries In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 95–104. ACM.

Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489.

Yao K, Zweig G, Hwang MY, Shi Y, Yu D (2013) Recurrent neural networks for language understanding. In *Interspeech*, pp. 2524–2528.

Yu Z, Scherer S, Devault D, Gratch J, Stratou G, Morency LP, Cassell J (2013) Multimodal prediction of psychological disorders: Learning verbal and nonverbal commonalities in adjacency pairs In *Semdial 2013 DialDam: Proceedings of the 17th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 160–169.

Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .

Zhang Z, Luo P, Loy CC, Tang X (2014) Facial landmark detection by deep multi-task learning In *European conference on computer vision*, pp. 94–108. Springer.

Zhong L, Zhong Z, Zhao Z, Wang S, Ashley KD, Grabmair M (2019) Automatic summarization of legal decisions using iterative masking of predictive sentences In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 163–172.