SEVERITY MEASURES FOR ASSESSING ERROR IN AUTOMATIC SPEECH RECOGNITION

by

Ryan Whetten

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Computer Science

Boise State University

May 2023

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Ryan Whetten

Thesis Title: Severity Measures for Assessing Error in Automatic Speech Recognition

Date of Final Oral Examination: 01 May 2023

The following individuals read and discussed the thesis submitted by student Ryan Whetten, and they evaluated the presentation and response to questions during the final oral e xamination. They found that the student passed the final oral examination.

Casey Kennington, Ph.D.	Chair, Supervisory Committee
Tim Andersen, Ph.D.	Member, Supervisory Committee
Michael Ekstrand, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Casey Kennington, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

dedicated to my brother, Andrew Whetten

ACKNOWLEDGMENTS

I would first like to express gratitude to my advisor, Casey Kennington, who has been an example to me throughout my Master's degree. Through many meetings, emails, lectures and conversations Casey has shaped me for the better, not only as my teacher and advisor, but as great example outside of the university campus as well. Casey helped me increase my understanding in natural language processing, turning me from an ambitious but clueless undergraduate, into someone who can understand and begin to conduct research in the field.

I would also like to thank Tim Andersen and Michael Ekstrand for their time and effort in being on my committee and for being my principal teachers though the past year and 8 months. All of the members of my committee have played a role in teaching me how to think deeper, how to learn independently and how to conduct research.

I am extremely grateful for Kavan Hess, Rachael Powell and Ryan Roper for taking time out of medical school to help me with this research.

Lastly, I would like to thank my family especially to my wife and brother, who have been my cheering squad and also my fuel for ideas and my inspiration to keep going. In fact, it was during a conversation with my brother when idea that turned into this thesis was born.

ABSTRACT

A common metric for evaluating Automatic Speech Recognition (ASR) is Word Error Rate (WER) which solely takes into account discrepancies at the word-level. Although WER is useful, it is not guaranteed to correlate well with intelligibility or performance on downstream tasks that make use of ASR. Meaningful assessment of ASR mistakes becomes even more important in high-stake scenarios such as health-care. I propose 2 general measures to evaluate the quality or severity of mistakes made by ASR systems, one based on sentiment analysis and another based on text embeddings. Both have the potential to overcome the limitations of WER. I evaluate these measures on simulated patient-doctor conversations. Measures of severity based on sentiment ratings and text embeddings correlate with human ratings of severity. Measures based on text embeddings have the capability to predict human severity ratings better than WER. These measures are used in metrics in the overall evaluation of 5 ASR engines alongside WER. Results show that these metrics capture characteristics of ASR errors that WER does not. Furthermore, I train an ASR system using severity as a penalty in the loss function and demonstrate the potential for using severity not only in the evaluation, but in the development of ASR. Advantages and limitations of this methodology are analyzed and discussed.

TABLE OF CONTENTS

ABSTRACT v

LIST OF TABLES viii

LIST OF FIGURES x

1 Introduction 1

2 Thesis Statement 5

3 Background 6

3.1 Sentiment Analysis and Embeddings 6

3.2 ASR Engines 9

4 Data 11

4.1 Data Collection 11

4.2 Rater Credentials 12

4.3 Data Validation: Do Raters Agree? 13

5 Methods 15

5.1 Experiment 1: Testing Severity Scores 15

5.1.1 From Sentiment Analysis and Embeddings to Severity Scores . . 16

5.1.2 Results 17

5.2 Experiment 2: Using Metrics To Evaluate ASR 20

5.2.1 Metric 1: MAE of Difference in Sentiment 21
5.2.2 Metric 2: MSE of Difference in Sentiment 21
5.2.3 Metric 3: Sentence Similarity using Language Models 22
5.2.4 Results 23
5.2.5 Advantages and Limitations 24
5.3 Experiment 3: Using Severity To Improve ASR 27
5.3.1 Adding Severity into Loss Function 28
5.3.2 Model 30
5.3.3 Data and Training Regime 31
5.3.4 Results 32
6 Conclusion 33
6.1 Discussion: Implications and Future Work 33
6.2 Summary 34
REFERENCES 36

LIST OF TABLES

4.1 Inner-annotator agreement confusion matrix. 14

4.2 Kendall's correlation coefficient 14

5.1 Correlation between human rating of severity to WER and various severity measures. The three italicized are severity scores based on sentiment analysis, the four bold are severity scores based on sentence embeddings. The sentence embedding scores have the highest correlation with human ratings of severity. 17

5.2 Examples of the sentiment analyzer, FLAIR, giving a high rating (close to 2) for opposites and negation. More examples are shown in 5.5 18

5.3 Mean accuracy of Ordinal Logistic Regression Models with 10-fold cross validation. All the models based on text embeddings have a higher mean accuracy than WER. 19

5.4 Results of Experiment 2. The top row shows each of the 5 ASR engines. The following section shows the WER. The labels in the first column that end in mae and mse are the mean absolute error and the mean squared error of the difference in sentiment scores respectively. The last for rows are the average cosine distance. 23

5.5 Examples of *severe* errors. The first 6 and second groups of 6 are based on sentiment and text embeddings respectively while WER is kept below 0.5. The last 6 are based on WER whie cosine distance of text embeddings is kept below 0.5. 25

5.6 Percent Increase in performance from base model to CTC-by-Cos model on both training and validation datasets. Severity is measured by the average cosine distance proposed in Section 5.3.3. 32

LIST OF FIGURES

1.1 WER calculations for the “I love you” example. S, D, I represent the number of substitutions, deletions, and insertions to go from the ground-truth transcription to the output of an ASR. N represents the total number of words in the ground-truth transcription. 2

4.1 Image showing the instructions given to raters and a few example pairs of sentences with the correct transcription on the left, the output of an ASR engine in the middle and the human rating of severity on the right. 12

5.1 Graphs comparing human ratings of severity (x-axis) to WER and two severity ratings one base on sentiment scores and the other based on sentence embeddings (y-axis). Note that a fair number of errors with a high human severity rating have a relatively low WER. 19

5.2 Main components of model implemented based on DeepSpeech2. 31

CHAPTER 1

INTRODUCTION

Automatic Speech Recognition (ASR) is the task of processing human speech into text. ASR has drastically improved over the past decade and has revolutionized the way many people interact with computers using applications such as voice search, dictation, and virtual assistants (e.g. Apple's Siri, Samsung's Bixby, Google Now) [56, 2]. It is common practice to evaluate ASR systems by calculating the word error rate (WER). The WER can be calculated by counting the number of words that need to be substituted (S), deleted (D), and inserted (I) to go from a ground-truth human transcription to the output of an ASR. This count is then divided the total number of words in the ground-truth transcription (N) [32]. WER is often written as $(S + I + D)/N$. Essentially, WER treats each discrepancy between the ground-truth transcription and the output from the ASR **equally**.

However, one issue is that not all ASR errors are equal. As an example, take the sentence "I love you," and suppose an ASR system produces "I loathe you." This would result in a WER of 0.33. Now let us suppose another ASR system predicts "I luv you." This too would result in a WER of 0.33 (see Figure 1.1). Although the WERs are equal, compared to the ground-truth "I love you," the mistake "luv" is arguably less severe than the mistake "loathe," which gives off the opposite meaning from the ground-truth sentence. This example, with two different errors and the same

WER, shows the way a human might perceive the severity of an error in transcription will not always be inline with WER.

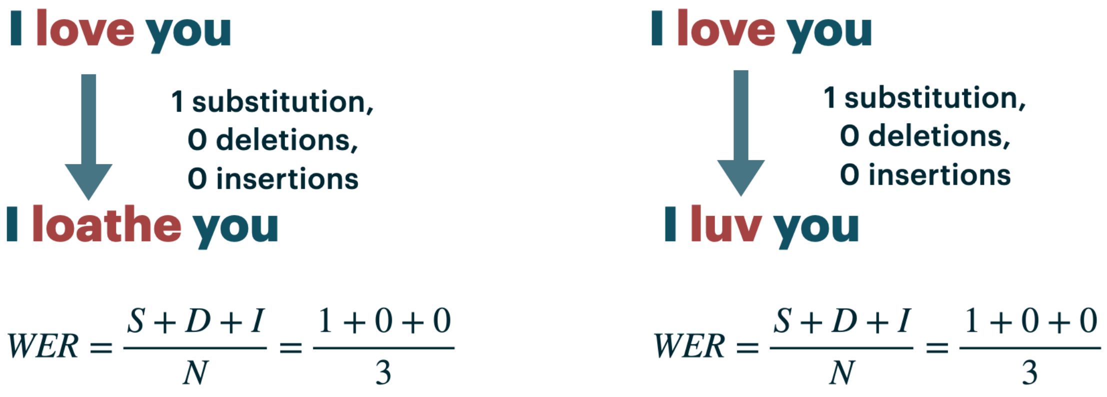


Figure 1.1: WER calculations for the "I love you" example. S, D, I represent the number of substitutions, deletions, and insertions to go from the ground-truth transcription to the output of an ASR. N represents the total number of words in the ground-truth transcription.

Researchers have studied the way humans perceive and rank ASR errors. They found that there was more consensus among raters in the extrema (i.e. the most and least severe errors) [37]. This consensus at the extrema suggests that there potentially exists loose patterns or some methodology that humans use to rate the seriousness of an error in transcription. Furthermore, when comparing edit distances, or the minimum number of operations required to transform one string into another, to the severity of the errors in transcription, they found that there were many serious errors that had a relatively low edit distance. This is in agreement with other studies, which similarly show that WER is not always well correlated with intelligibility or performance in a given downstream task, such as in natural language understanding (NLU) or named entity recognition (NER) [18, 54, 48].

Being able to understand the seriousness, or severity, of ASR errors becomes even more critical in high-stake scenarios such as healthcare. ASR has been used in healthcare since the 1970's and has proven to be beneficial, reducing the costs and time need in reporting [26]. In healthcare research, transcriptions are used in a wide variety of tasks such as in the automated detection of dementia [14], in estimating scores of standardized cognitive health screening tests [15], and in in the prediction and explanation of diagnosis [41]. All of these works operate on the basis of having an intelligible and accurate transcript. The purpose of this research is to develop a method for systematically measuring and understanding the quality of ASR systems, especially in high-stake settings like healthcare, that goes beyond WER by looking at the difference in meaning between the ground-truth and ASR output.

Needless to say, in high-stake environments like healthcare, without a useful measurement to understand the severity and potential impact of errors in transcriptions, it becomes hard to evaluate the quality of an ASR engine, and consequently, find key areas of improvement. In this research, I propose two methods for the automatic rating of the severity of errors in ASR transcriptions by using 1) the difference in sentiment ratings and 2) cosine distances between text embeddings of the output of the ASR and the ground-truth human transcription of a given audio. Sentiment ratings will capture the polarity of a given text and will act as a simplistic measure for capturing the *meaning* of that text. Text embedding have the potential to better capture the *meaning* of a given text and the cosine-similarity of text embeddings is a common method to compare the semantics of text [47]. By using the cosine distance, $1 - \text{cos-similarity}$, we get a value where, like WER, the higher the value the farther apart the ground-truth is from the ASR output. I test the viability and usefulness of these methods on simulated medical data. I do this by 1) comparing the results of these measures to human severity labels (Section 5.1), and by 2) incorporating these measures in to metrics for the overall evaluation of ASR engines (Section 5.2). The results of these metrics on 5 different ASR architectures are compared.

The results show there is reliable consensus among human raters based on Fleiss' Kappa, Cohen' Kappa, and Kendall's correlation measures [17, 9, 27]. The difference in sentiment ratings correlates with human ratings of severity, but not as well as WER or the cosine distance of text embeddings. Text embeddings prove to be better predictors of human labels of severity than WER. This work also shows sentiment ratings, text embeddings, and WER capture different aspects of mistakes in transcriptions and shows there are advantages and limitations of each method. In the last experiment (Section 5.3), I demonstrate the potential for severity to be used in the development and improvement of ASR systems. I conclude by discussing future areas of research.

CHAPTER 2

THESIS STATEMENT

Before developing a method for estimating the severity of ASR errors using sentiment analysis and/or text embeddings, I first seek to answer the following question: to what extent does the information captured by sentiment analyzers and/or text embeddings correlate with human assessment of the severity of ASR errors in a healthcare setting?

If this first question reveals that sentiment analyzers and/or text embeddings do correlate with human assessment to some degree, then we have evidence that we could use sentiment analysis or text embeddings in the systematic rating of ASR errors. This leads to the following questions: can we use a sentiment analyzer and/or text embeddings to develop a useful automatic estimator for the rating of the severity of ASR errors? If so, can this be used in the overall evaluation of an ASR system? How would this evaluation compare to WER? What are the advantages and limitations of these methods? And lastly, could these be used in the development of ASR systems?

I hypothesize that sentiment analysis and text embeddings do capture enough information to develop an automatic method to rate the gravity of ASR errors in a healthcare setting. I also predict that we can also use these methods in conjunction with WER to better evaluate the overall performance of an ASR engine.

CHAPTER 3

BACKGROUND

In this section, I give a brief overview of sentiment analysis and textual embeddings, including the specific sentiment analyzers and models for textual embeddings used in this work. Then, I introduce the ASR engines that I use for the experiments.

3.1 Sentiment Analysis and Embeddings

When it comes to understanding and automatically rating the seriousness of errors in ASR, one needs to have a method for systematically analyzing the difference in *meaning* between two phrases or sentences. While philosophically what a body of text truly means is a difficult question to answer, one simple way of capturing some essence of the *meaning* of an utterance is to do a sentiment analysis on the output of an ASR engine and the ground-truth transcription.

In the field of Natural Language Processing (NLP), sentiment analysis is the task of detecting the attitude, emotions, or polarity of a given text. It is common for these algorithms to take in a string as input and output a prediction from -1 to 1 based on how negative or positive the text is. Because these algorithms can vary and have their own limitations, I use 3 different sentiment analyzers from 3 different widely-used NLP libraries NLTK, FLAIR, and TextBlob (TB) [24, 1, 36]. This is a naive method of capturing the *meaning* of a given text because, clearly two texts can have different meanings yet both have the same sentiment. Although perhaps overly simplistic, the purposes of using sentiment is to create somewhat a baseline measure that captures something besides discrepancies in spelling and to test how well sentiment ratings can perform.

Another common method for capturing the *meaning* of natural language is to use text embeddings. Generating word embeddings is the process of converting individual words into n-dimensional vectors, usually for the purposes of converting text into something that can be processed by Machine Learning or Deep Learning algorithms. There are a variety of methods for embedding words that range from simple rule-based methods to more complicated methods that involve machine learning techniques [38, 43, 44]. Similarly, methods have been developed for embedding more than just single words [31, 47]. Whether embedding individual words or entire sentences, generally, with good embeddings, the more semantically similar words or phrases are, the closer they should be in the n-dimensional vector-space [38, 39].

For this project I use 4 readily available pre-trained models provided by Sentence-Transformers[1] to compute sentence embeddings. Below is a brief description of each one.

bert-base-nli-mean-tokens (BertNLI) is a modification of the pre-trained BERT model that "use(s) siamese and triplet network structures to derive semantically meaningful sentence embeddings" [47]. Using raw BERT for large scale semantic searching or comparison is computationally expensive, however with this methodology big networks can be fine tuned for semantic similarity using natural language inference

[1] https://www.sbert.net/docs/pretrained_models.html

data. All of the models are based on this same process of taking a big pre-trained language model and fine-tuning it on data for efficient semantic similarity.

all-MiniLM-L6-v2 (MiniLM) is based on Microsoft's MiniLM model [53]. For the base model, researchers developed a method of knowledge distillation called deep self-attention distillation, in which the purpose is to distil, or shrink down, a massive model, usually containing hundreds of millions of parameters, into a smaller model that generally maintains performance of the bigger model and can be more widely used. This base model was then fine-tuned for semantic similarity.

all-mpnet-base-v2 (MPNET) is based on Microsoft's MPNet model [51], which involves a combination of masked language modeling (a method of pre-training used in models like BERT [10]) and permuted language modeling in pre-training (a pre-training method used in XLR [55]), seeking to take advantages of both methods. This base model was then fine-tuned for semantic similarity.

all-distilroberta-v1 (DisRob) the last model I use, is based on DisilRoBERTa which is the follows the same process of distillation as DistilBERT [49], except with RoBERTa [34] as the base intead of BERT. The purpose of distillation of BERTA of RoBERTA is to shrink down the models size to increase speed and decrease memory requirements while keeping up performance. Like all the previous models, this base model was then fine tuned for semantic similarity.

3.2 ASR Engines

For this project I use five ASR engines for experimentation in order to collect and obtain results from a variety of architectures. I choose the following architectures because of there availability, performance and because they can be run locally (which means one would not have to deal with potential issues with sending sensitive data over the internet to a cloud ASR systems). In this section, I give a brief description of these five architectures.

Mozilla's DeepSpeech2 (DS2) is an implementation of [3]. In this architecture, Recurrent Neural Networks take in spetrograms from an audio file and are trained to output text[2].

Meta's Wav2Vec2 (W2V2) is a model proposed by [4]. Unlike DeepSpeech, this architecture operates directly on the raw audio data instead of spretrograms. The model is trained first in a semi-supervised method on many hours of unlabeled speech data and then is fine tuned on labeled data. This model is made easily accessible by HuggingFace [3].

CMU'S PocketSphinx (PS) is one of the lighter ASRs I use [23]. PS is a lightweight ASR that is a part of the open source speech recognition tool kit called the CMUSphinx Project. This model was trained on 1,600 utterances from the RM-1 speaker-independent training corpus. Unlike the previously mentioned models, PS does not use neural networks and is instead based on traditional methods of

[2] https://deepspeech.readthedocs.io/en/latest/index.html

[3] https://huggingface.co/docs/transformers/model_doc/wav2vec2

speech recognition by using Hidden Markov Models, language models, and phonetic dictionaries.[4]

Alpha Cephei's Vosk (with the vosk-model-en-us-0.22 model) is built using Kaldi [45], and like PS, uses an acoustic model, language model, and phonetic dictionary. However unlike PS, Vosk uses a neural network for the acoustic model part of the engine.[5]

OpenAI's Whisper unlike Wave2Vec2, uses a purely supervised method of training gathering 680K hours of transcribed content from the internet in 99 different languages [46]. Following other architectures such as DeepSpeech2, this model takes spectrograms of audio as input, but instead of Recurrent Neural Networks, this models uses an encoder-decoder Transformer architecture based on [52] with a variety of special tokens used to indicate which task is being performed (ex. transcription or translation). For my experiments, I use the base model[6] (consisting of 74 million parameters).

[4] `https://github.com/cmusphinx/pocketsphinx-python`

[5] `https://alphacephei.com/vosk/`

[6] https://huggingface.co/openai/whisper-base

CHAPTER 4

DATA

4.1 Data Collection

For the purposes of experimenting in a healthcare scenario, a dataset published in 2022 of simulated patient-physician medical interviews is used [13]. This dataset contains 272 audio files with transcripts. These files range from about 7 to 20 minutes or from 800 to 2200 words.

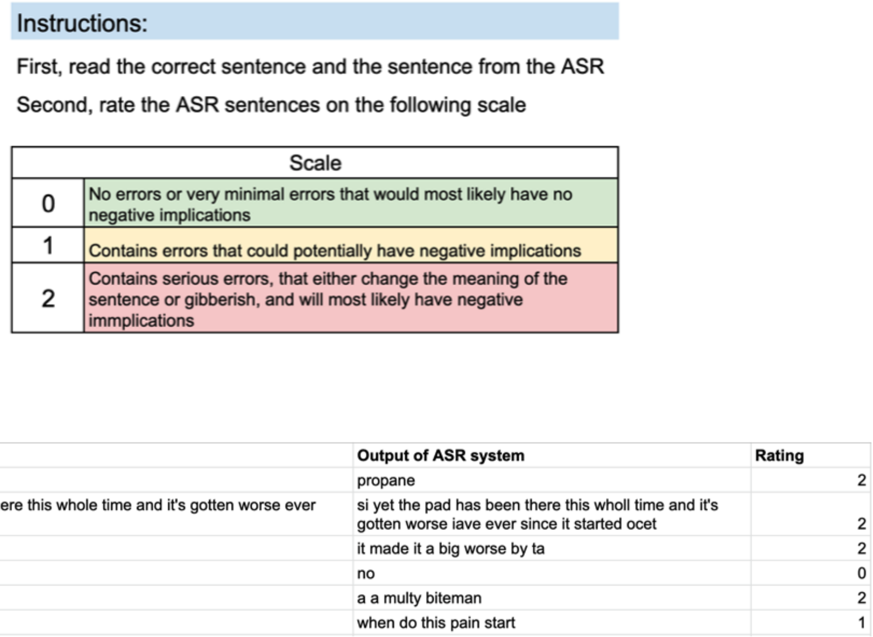
The files are split into non-silent intervals using librosa[1] setting the threshold of silence to 60 decibels. With a threshold of 60 decibels, the files split into over 39,600 non-silent intervals, which I will call utterances as each file contains a small utterance of speech. Of these, I take a sample of 110 utterences and run them through the ASR engines as well as manually obtain the transcription of the utterences from the corresponding transcription files that came with the dataset. Because these were run through five ASR engines, the result is a list of 550 pairs of transcripts where one comes from an ASR engine and the other is the ground-truth transcript. One of these *utterances* actually contained no speaking and was removed for a final total of 545 pairs.

150 of these pairs of transcripts were given to 3 medical school students who were asked to rate each pair with either 0, 1, or 2 (2 being a severe error, 1 being a not

[1] https://librosa.org/doc/main/generated/librosa.effects.split.html

so severe error, and 0 being a very minor error or perfect transcription). The exact instruction given and a few examples of the data are provided in Figure 4.1.

The pairs were normalized by removing speaker identification notes “P:” and “D:” for patients and doctors, making all letters lowercase, and by removing any special characters and punctuation except for apostrophes (as these could be important in distinguishing words like “its” and “it’s” or “they’re” and “their”).



Instructions:

First, read the correct sentence and the sentence from the ASR

Second, rate the ASR sentences on the following scale

Scale	
0	No errors or very minimal errors that would most likely have no negative implications
1	Contains errors that could potentially have negative implications
2	Contains serious errors, that either change the meaning of the sentence or gibberish, and will most likely have negative immplications

Correct	Output of ASR system	Rating
okay	propane	2
sorry yeah the pain has been there this whole time and it's gotten worse ever since it started	si yet the pad has been there this wholl time and it's gotten worse iave ever since it started ocet	2
it made it a bit worse but	it made it a big worse by ta	2
no	no	0
a multivitamin	a a multy biteman	2
when did this pain start	when do this pain start	1

Figure 4.1: Image showing the instructions given to raters and a few example pairs of sentences with the correct transcription on the left, the output of an ASR engine in the middle and the human rating of severity on the right.

4.2 Rater Credentials

All three raters are currently enrolled in a doctoral program at the Idaho College of Osteopathic Medicine (ICOM). Experience of the members includes medical research at locations such as the Mayo Clinic and the University of Utah, work as Spanish-English interrupter in medical clinics, work as anesthesia technician, and holding positions such as student representative on ICOM's research committee.

4.3 Data Validation: Do Raters Agree?

Previous work suggests that the severity of errors in transcription is a difficult task where there is not very good consensus among raters [37]. Prior to developing a measure that rates errors in the same way a human would, it first needs to be shown that humans do have some methodology or consistency amongst each other when it comes to rating the severity of errors.

Following the evaluation metrics used in [37], I use Cohen's Kappa [9] and Fleiss' Kappa [17], to measure at inner-annotator agreement. However, these metrics do not take in to account that the data is ordinal (i.e. a discrepancy in ratings of values 0 and 1 is treated the exact same as a discrepancy in values of 0 and 2 even though the latter discrepancy is greater than the former [12]). Therefore, since the nature of these ratings is ordinal, I also look at the Kendall's rank correlation coefficient [27] to measure the quality of the ordinal association between two given raters.

I calculate a Fleiss' Kappa values of 0.452 and Cohen's Kappa scores that range from 0.420 to 0.567, which shows moderate agreement between raters (see Table 4.1). The Kendall's correlation coefficient between raters indicated a strong correlation between rater ranging from 0.662 to 0.727 (see Table 4.2). Considering the subjectivity of the task, the moderate Kappa values and high correlation values suggests that there is reliable consistency among raters.

Table 4.1: Inner-annotator agreement confusion matrix.

	Rater 1	Rater 2	Rater 3
Rater 1	-	0.416	0.567
Rater 2	0.416	-	0.440
Rater 3	0.567	0.440	-

Table 4.2: Kendall's correlation coefficient

	Rater 1	Rater 2	Rater 3
Rater 1	-	0.727	0.718
Rater 2	0.727	-	0.662
Rater 3	0.718	0.662	-

CHAPTER 5

METHODS

This chapter describes the experiments and methodology used to test the hypotheses described in Chapter 2.

5.1 Experiment 1: Testing Severity Scores

In this experiment, the goal is to test if sentiment analyzers and/or text embeddings can rate errors similarly to the way a human would in a healthcare setting. I use the 150 utterances that have human labels for severity described in Section 4.1 and since there is decent agreement, I use the mode of the ratings as a target.

I calculate the WER and various *severity scores* (defined below in Section 5.1.1) using the sentiment analyzers and the language models for text embeddings described in Section 3.1. I compare the severity scores with each other by measuring the correlations between the severity scores and the mode human rating. A high correlation between these *severity scores* and the human ratings supports the hypothesis that sentiment analysis and/or text embeddings capture enough information to be used to evaluate the severity of ASR errors in a healthcare setting.

To further evaluate the usefulness of these *severity scores*, I create multiple Ordinal Logistic Regression models with a single *severity score* as a independent variable in each model and the mode human rating as the target variable and compare the performance of the models. To compare, I perform 10-fold cross validation for each model and look at the mean accuracy on the test data and compare them with each other and with a majority classifier.

5.1.1 From Sentiment Analysis and Embeddings to Severity Scores

In this section I explain how I calculate the various *severity scores* using sentiment analyzers and text embeddings.

When rating the severity of errors as a human, we could try to objectively look at the *difference in meaning* between the ground-truth and the output of an ASR engine. One could imagine this process potentially involving using some model in the brain where *love* is closer to *luv* than *loathe* in meaning, and thus *luv* would be rated as a less severe error. I seek to mimic this process using sentiment analyzers and text embeddings.

Given a sentiment analyzer $s(x)$ that outputs a value between -1 and 1 (such as NLTK's SentimentIntensityAnalyzer [6]), we can try to mimic this process by taking the absolute value of the difference in sentiment and use this as model to represent the *difference in meaning* or severity. This can be expressed by the following:

$$\text{Severity}(x, y) = |s(x) - s(y)|$$

where x and y are a pair of ground-truth and ASR output. This would result in a rating on the range [0, 2], where 0 would be two phrases that have the exact same sentiment rating, in other words a non-severe error, if any, and a rating of 2 would represent the most severe error possible due to having sentiments and polar extremes.

Following the same logic with text embeddings, knowing that closer embeddings should be semantically closer, we can represent the *difference in meaning* as the dissimilarity between them, in other, words one minus the cosine of their embeddings.

$$\text{Severity}(x, y) = 1 - cosine(x, y)$$

This would result in a rating on the range [-1, 1]. However, it is common practice to bound the vectors in the positive space which would result in range of [0, 1]. Because we are looking at the dissimilarity, and value of 0 would represent two strings that are the same and a value close to 1 would represent a two strings that are very different semantically.

5.1.2 Results

The correlations shown in Table 5.1 show WER has a correlation with human ratings of severity of 0.43. All the severity scores based on text embeddings correlated with human ratings better than WER, with an increase in correlation ranging from 28% to 36% above WER. In contrast, all of the severity scores based on sentiment were less correlated than WER. The sentiment analyzer from FLAIR correlates the most with human ratings with a value of 0.34 (see Table 5.1). The results show that text embeddings can be better suited for the automatic evaluation of severity in ASR errors than WER.

Table 5.1: Correlation between human rating of severity to WER and various severity measures. The three italicized are severity scores based on sentiment analysis, the four bold are severity scores based on sentence embeddings. The sentence embedding scores have the highest correlation with human ratings of severity.

WER	*NLTK*	*FLAIR*	*TB*	**MiniLM**	**BertNLI**	**MPNET**	**DisRob**
0.43	0.29	0.34	0.29	0.55	0.53	0.56	0.59

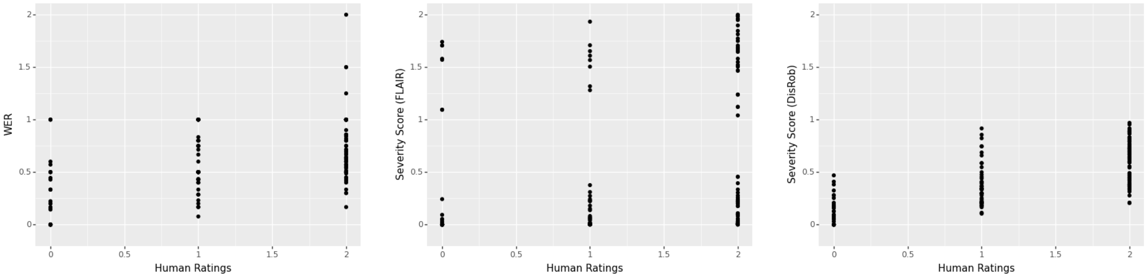
Plots comparing WER, FLAIR, and all-distilroberta-v1 (DisRob) to human ratings are in Figure 5.1. Subplot 5.1c, which compares human ratings to DisRob, shows that embeddings do the best job of clustering ASR errors with the same human rating together. In other words generally, the most severe errors (with a human rating of 2) are pushed towards 1 (the max for this measure), the medium-severe errors (with a human rating of 1) are clustered around 0.5, and the lest severe errors (with a human rating of 0) are between 0 and 0.5.

In contrast, the WERs shown in subplot 5.1a are more spread out, having some severe errors with relatively low WER and some non-severe errors with a relatively high WER.

The middle plot, subplot 5.1b, shows severe errors tend to be pushed to have a higher scores according to FLAIR sentiment analyzer, but, like WER, FLAIR rates some severe errors with a low score. Looking deeper, severity scores based on sentiment tended to do well on detecting opposites or negations such as “love” and “loathe” or “not laying down” vs. “laying down” which is in line with their purpose (i.e. to detect polarity). Due do this, I believe there is still some potential in using sentiment to rate severity, especially for detecting opposites or misses in negation as in the “love” and “not laying down” examples, but based on these results, it will most likely not be as versatile as text embeddings.

Table 5.2: Examples of the sentiment analyzer, FLAIR, giving a high rating (close to 2) for opposites and negation. More examples are shown in 5.5

Ground-Truth ASR output	FLAIR	ASR
um not laying down helps laying down help	1.943	DeepSpeech2
i love you i loathe you	1.989	Toy example



(a) WER

(b) Sentiment (Flair)

(c) Embeddings (DisRob)

Figure 5.1: Graphs comparing human ratings of severity (x-axis) to WER and two severity ratings one base on sentiment scores and the other based on sentence embeddings (y-axis). Note that a fair number of errors with a high human severity rating have a relatively low WER.

Using Severity Scores as a Predictor

For results of the Ordinal Logist Regression models, I create models using all the same severity measures previously mentioned, as well as a majority classify as a baseline. The worst performing model is the model using the scores based on FLAIR with a mean accuracy of 50% which is equivalent to the baseline. The model based on WER was the next best with a mean accuracy of 62%. All of the models based on text embedding scores perform better than the WER model with the best model having a mean accuracy of 70.67%. The results are summarized in Table 5.3.

Table 5.3: Mean accuracy of Ordinal Logistic Regression Models with 10-fold cross validation. All the models based on text embeddings have a higher mean accuracy than WER.

Maj. Class.	WER	FLAIR	MiniLM	BertNLI	MPNET	DisRob
50.00	62.00	50.00	66.00	63.33	66.67	**70.67**

The results from the Ordinal Logistic Regression task further give evidence that text embeddings are better suited for the automatic evaluation of severity in ASR errors than WER.

In this experiment, I only study the correlation between the proposed severity scores and its corresponding human rating. However, it is common for WER to be averaged across all the utterances in a test dataset. The average WER becomes a single value that is used as a metric to gauge the overall performance of an ASR engine. Since the severity scores are well correlated with human labels, it is worth experimenting to see if these scores can be used to calculate a metric, in a similar manner to average WER, that could be used in gauging the performance of an ASR engine.

5.2 Experiment 2: Using Metrics To Evaluate ASR

This experiment demonstrates the potential usefulness of using the severity scores (from Experiment 1) in metrics for the overall evaluation and comparison of ASR systems. To test this, I propose three metrics (listed below, Section 5.2.1) and use each metric to evaluate and compare the performance of the five ASR engines from Section 3.2.

In order to do this evaluation, I take the 110 utterances that have been manually transcribed and run them through all five ASR engines. As mentioned in Section 4, one of the audio files did not have any speech in it and was removed. This results in a total of $109 * 5 = 545$ transcriptions. As before, I compute the various severity scores for each ASR output, ground-truth pair. These severity scores are then used as input into the metrics described in the following section.

These metrics are compared with WER and with each other across the five ASR engines to evaluate how these metrics behave.

5.2.1 Metric 1: MAE of Difference in Sentiment

The first metric I propose is the mean absolute error of the differences in sentiment (MAE-DS). Formally, given a sentiment analyzer $s(x)$ that outputs a value between -1 and 1 (such as NLTK's SentimentIntensityAnalyzer [6, 24]), we can express the MAE-DS in the formula below:

$$\frac{1}{n}\sum_{x,y\in C}|s(x)-s(y)|$$

where C is a corpus of pairs of ground-truth transcriptions and ASR predictions and n is the number pairs. x and y are a set of ground-truth and predicted utterances from C.

The output of this metric will range from 0 to 2, and will be easy to interpret. For example, an MAE-DS 0.5, indicates that, in terms of sentiment, the ASR's output is, on average, 0.5 off of the ground-truth.

5.2.2 Metric 2: MSE of Difference in Sentiment

The second metric I propose is the mean squared error of the differences in sentiment (MSE-DS). Similar to the first metric, given a sentiment analyzer $s(x)$ we can write this in the formula below:

$$\frac{1}{n}\sum_{x,y\in C}(s(x)-s(y))^2$$

where, C is a corpus of pairs of ground-truth transcriptions and ASR predictions and n is the number pairs. x and y are a set of ground-truth and predicted utterances from C.

The range of this metric is from 0 to 4. The potential usefulness of this metric lies in the fact the MSE is more sensitive to outliers than MAE. As a result, this will penalize more heavily the ASR errors that have a greater distance in sentiment from the ground-truth.

5.2.3 Metric 3: Sentence Similarity using Language Models

The third metric I propose is based on text embeddings genrerated by language models. In this case, I calculate the cosine similarity between the embeddings of the ground-truth and ASR output as a representation of how different the sentences are in meaning. In the model that I will use, the cosine similarity of embeddings will generally range from 0 to 1 where one is the exact same sentence and values closer to 0 being semantically distant from the target sentence [47].

With this, I propose using the mean of the cosine distance, or one minus the cosine similarity. This can be written with the following formula:

$$\frac{1}{n}\sum_{x,y\in C} 1 - cosine(x,y)$$

where C is a corpus of pairs of ground-truth transcriptions and ASR predictions and n is the number pairs. x and y are a set of embeddings of a given ground-truth and predicted utterances from C.

The potential usefulness of this metric is that text embeddings can capture more information than sentiment ratings and that the language models that generate the embeddings can also be fine tuned to a given dataset (i.e. in the medical field, one could make it so that names of medicines and diagnoses are farther from each other). However, no fine tuning is done in this work. Another potential advantage of this method it that this is very similar to the Cosine Loss function, and could potentially be easily incorporated into the loss function as a penalty for neural network training.

5.2.4 Results

The results are summarized in Table 5.4. For all the metrics, the lower the value the better. Generally results are consistent, no matter which metric we use, the majority show that Whisper has the best performance followed by Vosk. Following these in performance are DeepSpeech2 (DS2) and Wav2Vec2 (W2V2). The ASR with the lowest performance is PocketShpinx (PS). Despite the metrics agreeing for the most part, going into more depth and studying to what extent these metrics agree and where the metrics disagree, one can gain useful insights about what information the proposed metrics are capturing.

Table 5.4: Results of Experiment 2. The top row shows each of the 5 ASR engines. The following section shows the WER. The labels in the first column that end in mae and mse are the mean absolute error and the mean squared error of the difference in sentiment scores respectively. The last for rows are the average cosine distance.

Base Measure	Metric	DS2	PS	Vosk	Whis.	W2V2
Edit Distance	WER	0.482	0.910	0.307	**0.273**	0.525
Sentiment	NLTK_mae	0.127	0.241	0.062	**0.056**	0.127
	FLAIR_mae	0.620	0.700	0.324	**0.322**	0.516
	TB_mae	0.111	0.181	0.050	**0.029**	0.120
Sentiment	NLTK_mse	0.057	0.141	0.022	**0.020**	0.048
	FLAIR_mse	0.981	1.132	**0.459**	0.473	0.788
	TB_mse	0.051	0.086	0.026	**0.010**	0.044
Cosine Distance	MiniLM	0.361	0.649	0.171	**0.153**	0.403
	BertNLI	0.188	0.398	**0.079**	0.093	0.181
	MPNET	0.400	0.688	0.193	**0.180**	0.400
	DisRob	0.388	0.676	0.189	**0.172**	0.406

From Vosk to Whisper there is a percent decrease in WER of about 11.07%. The average percent decrease in the cosine distance over the 4 language models is quite small at 2.13%. In another example, the percent decrease in WER from DeepSpeech2 to Vosk is 36.31% while the average percent decrease cosine distance over the 4 language models is greater at 53.41%. The differences in these percentages show that the rate of improvement in the *severity* (cosine distance) is not necessarily related to rate of improvement in WER (i.e. one metric can improve greatly while the other not so much and vice versa).

To further demonstrate the differences of these metrics, I look at specific examples where WER and measures of severity disagree. I do this by analyzing the most *severe* errors given a certain measure where another measure is kept relatively low. I first look at the most *severe* according to FLAIR sentiment scores while keeping WER below 0.5 (examples from this are shown in the first 6 rows in Table 5.5). I then look at the most *severe* according to cosine distance while still keeping WER below 0.5 (shown in the middle group of 6 in Table 5.5). Finally, I look at the most *severe* according to WER while keeping the cosine distance below 0.5 (the last 6 examples in Table 5.5). These edge case examples show potential advantages and limitations of WER, sentiment scores, and scores based on text embeddings.

5.2.5 Advantages and Limitations

All of the following examples in this section come from Table 5.5.

WER has the main advantage of being simple and consistent; it is just the edit distance normalized by the total number of words. There are not multiple models like how there are various language models for sentiment analysis and for generating text embeddings. The main limitation of WER is that, because it is based on edit distance and not any *understanding* or model of the language, there are severe errors

Table 5.5: Examples of *severe* errors. The first 6 and second groups of 6 are based on sentiment and text embeddings respectively while WER is kept below 0.5. The last 6 are based on WER whie cosine distance of text embeddings is kept below 0.5.

Ground-Truth ASR output	FLAIR	MiniLM	WER	ASR
uh i smoke about a pack a day uh smoke about a pack of day	1.929	0.104	0.250	Whis.
and how often do you use crystal meth and how often do you use crystal mud	1.858	0.371	0.125	Whis.
ok sounds like a a pretty stressful job and like a pretty stressful job	1.850	0.298	0.375	DS2
uhm it started last night and it started last night	1.707	0.138	0.200	W2V2
what they did for your heart attack what they did for your herd attack	1.617	0.546	0.143	W2V2
any previous surgeries any previous surgery	1.580	0.111	0.333	DS2
nothing has seemed to make it any... dorthins seemed to make him any...	0.003	0.692	0.364	W2V2
what they did for your heart attack what they did for your herd attack	1.617	0.546	0.143	W2V2
and how often do you use crystal meth and how often do you use for sunlight	0.010	0.512	0.250	Vosk
that you're experiencing some chest pain that you're experiencing some testing	0.049	0.469	0.333	Whis.
about the same ok and has it gotten... the same moqe and has it gotten more...	0.028	0.461	0.200	W2V2
that you're experiencing some chest pain that you're experiencing some chatting	1.889	0.456	0.333	Vosk
ok okay	1.094	0.061	1.000	DS2
a multivitamin a multi vitamin	0.000	0.150	1.000	DS2
my parents our friends	0.005	0.370	1.00	PS
i've tried uh i have tried add	1.733	0.451	1.000	Vosk
uh thirty eight degrees 38 degrees	0.161	0.177	0.750	Whis.
uh thirty eight degrees the degrees	0.007	0.324	0.750	DS2

that have a relatively low WER, and vice versa, there are non-severe errors that have a high WER such as *a multivitamin* vs. *a multi vitamin.*

Sentiment has strong limitations due to the fact that the algorithms used to generate sentiment scores are designed to only capture how positive or negative a given text is. Sentiment scores can also be highly sensitive to misses in disfluencies like *um* or *uh*. This is highlighted in the example, *uhm it started last night* vs *and it started last night*, where there was a strong difference in sentiment of 1.707. This can be an advantage or a limitation depending on the scenario. Many ASR engines overlook disfluencies, but, for example in human robot interactions or spoken dialogue systems, disfluencies can be vital to understanding and performance [5, 8]. Besides human computer interaction, disfluencies in transcripts can also be critical in a healthcare setting where differences in speech fluency are used as predictors of dementia status [14, 35, 40].

There is the also the limitation on the performance of the model, where the model incorrectly classifies the sentiment. In the examples *any previous surgeries* vs. *any previous surgery* or *uh i smoke about a pack a day* vs. *uh smoke about a pack of day*, there is a high difference in sentiment yet the only difference is in missing the pronoun *i* or the plural of *surgery*, which should not affect sentiment greatly.

However despite these limitations, sentiment is able to catch severe errors where the WER is relatively low. In the example where *crystal meth* becomes *crystal mud* or where *chest pain* becoming *chatting* the WER is 0.125 and 0.333 respectively but the difference in sentiment is very high at 1.858 and 1.889 respectively.

Text embeddings are limited by the performance of the model, like sentiment, yet, can capture more than just polarity of a given text. Knowing that many of these models are trained in a self-supervised manor using the context in the training text, we can imagine how the embeddings of *my parents* and *our friends* could be similar. Both of these phrases could occur in with similar surrounding text; they have the same grammatical structure (a possessive adjective followed by a noun) and parents and friends are both human relationships.

Another limitation on these models is the amount of text they can handle. Anything above the model's limit gets truncated and, consequently, loses the meaning of truncated text. Although there are many short utterances in ASR training data. The character limitation on these models could affect performance on longer utterances.

Despite their limitations, text embeddings were able to capture well the differences in *meaning*. Text embeddings were able to give a high score to the examples where *crystal meth* becomes *for sunlight* and where *chest pain* becomes *testing* when WER and sentiment scores were relatively low. Text embeddings were also able to give low ratings for different writings of the word *okay* and numbers (*ok* vs. *okay*, or *uh thirty eight degrees* vs. *38 degrees*) when WER were high.

5.3 Experiment 3: Using Severity To Improve ASR

Up to this point results show that 1) there is reliable consistency among human raters, 2) the cosine distance of text embeddings correlates better with human labels of severity than WER, and 3) using sentiment or text embeddings in a metric for the overall evaluation of ASR captures different information than WER. With these results established, the purpose of this experiment is to test if an automatic measure of *severity* can be used in more than just the evaluation of ASR, but in the development as well.

Previous work done in the study of ASR errors involves approaches to automatically detect errors using word and text embeddings (and even other features such as acoustic/prosodic features), [19, 20, 21] and to automatically repair errors in specific cases (such as in certain homophones in French) [11]. However, instead of ASR error detection or repair which happens post-prediction, my approach will be to include severity into the making of an ASR system in an attempt to reduce the number of errors (measured by WER) and to reduce the overall severity of the errors produced (measured by the average cosine distance proposed in Section 5.3.3). To do this, I incorporate *severity* into the loss function during training of an ASR. The exact methodology to incorporate *severity* into the loss function, the model used, and the experiments I perform are described in the following sections.

5.3.1 Adding Severity into Loss Function

It is common for ASR engines that involve neural networks to be trained using the Connectionist Temporal Classification (CTC) loss function [22]. This algorithm allows one to work with data where both inputs and outputs can vary in length such as in handwriting recognition and speech recognition. CTC sums over the probabilities of all possible alignments between the input and the output. Naturally, this can be quite expensive. To overcome this, the CTC algorithm takes advantage of dynamic programming methods to efficiently compute the probability of each output. In other words, given and input of audio X and a ground-truth transcript Y, CTC can calculate efficiently $p(Y|X)$.

When training a neural network we ideally want to maximize the likelihood of the ground-truth, Y, given the corresponding audio, X, to be as close to 1 as possible. Since the likelihood can be extremely small, it is common for the models parameters to be tuned tuned minimize the negative log-likelihood, $\sum_{(X,Y)\in D} -log\ p(Y|X)$, where D is a given training set.

To incorporate *severity* into the loss function, the cosine distance is used as a weight in the loss function. To calculate this weight, w, the cosine distance is limited in the range from a near-zero number, 1.0×10^{-7}, to 1. This is shown in Equation 5.1, where W is the weight that represents the *severity* between the ground-truth, Y_{truth}, and the output of the ASR, Y_{pred}. This weight is multiplied by the CTC loss value to get the final loss (Equation 5.2).

This results in a function where the original CTC loss is scaled down, along with the gradients of the neural network, based on the semantic similarity between the ground-truth and ASR output.

$$\mathrm{w} = 1 - max(1.0 \times 10^{-7}, cos(Y_{truth}, Y_{pred})) \tag{5.1}$$

$$L = w * \mathrm{CTC} \tag{5.2}$$

I will refer to this proposed loss function as a CTC-by-Cosine loss function due to its use of CTC and cosine similarity. For this experiment I use the *all-MiniLM-L6-v* (MiniLM) to generate the embeddings for the ground-truth and ASR predictions.

5.3.2 Model

With a plan on how to incorporate severity into the development of an ASR system fixed, I now describe the model I use in this experiment.

The model is based on DeepSpeech2 [3], where the input is spectrogram from audio files and the output is the probability distribution of over a set of characters at each time step. The set of characters consists of all the letters of the English alphabet along with the following characters: apostrophe, questions mark, exclamation mark, and blank symbol.

The model starts with two 2D convolutional layers, both with 32 filters and batch normalization and goes through a ReLU activation function after each layer. The kernels for the convolutional layers are [11, 41] and [11, 21]. After the convolutions, there are five bidirectional gated recurrent layers (GRU) each with 512 units with a dropout layer with a rate of 0.5 after each recurrent layer except for the last one.

After the last recurrent layer there are two dense layers. The first one maintains the same size as the recurrent layers and passed through a ReLU activation function and a dropout layer with a rate of 0.5. The second dense layer is the output layer with softmax as the activation function. Adam is used for optimization with a learning rate of 1.0×10^{-4}. Figure 5.2 depicts the core components of this model.

This results in a model of about 26M parameters. This is relatively small compared to other ASR systems. For example, DeepSpeech2 has 38M parameters (about 1.5 times more parameters than the model used in this work), the base version of Wav2Vec2 has 95M parameters (about 3.7 times more parameters), and the base version of Whisper contains 74M parameters (about 1.2 times more parameters). However, the purpose of this experiment is not to achieve state of the art performance with a novel architecture, it is to test on a relatively small scale the plausibility of using severity in the development of an ASR system.

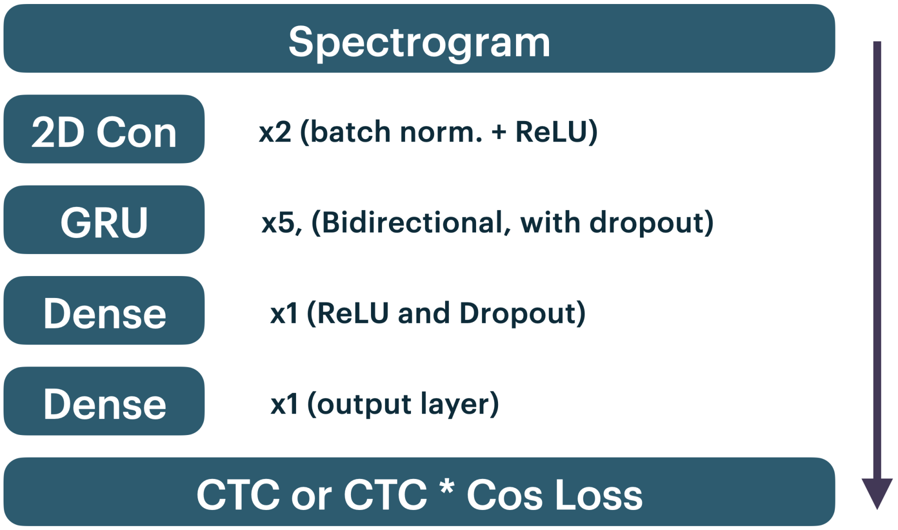


Figure 5.2: Main components of model implemented based on DeepSpeech2.

5.3.3 Data and Training Regime

Because the 109 transcribed utterances of the simulated patient doctor conversation files is insufficient to train an ASR engine from scratch, for this experiment I use the LJ Speech Dataset which consists of “13,100 short audio clips of a single speaker reading passages from 7 non-fiction books” [25].

I train 2 models on the first 90% percent of the data, withholding the last 10% for validation. The first model has the architecture described above and uses only the CTC loss function. This model will serve as the baseline. The second model uses the exact same architecture and training regime, but uses the CTC-by-Cosine loss function for the last 5 epochs.

Table 5.6: Percent Increase in performance from base model to CTC-by-Cos model on both training and validation datasets. Severity is measured by the average cosine distance proposed in Section 5.3.3.

	Train WER	Train Cos	Val WER	Val Cos
Base	0.058	0.051	0.268	0.249
CTC-by-Cos	0.008	0.006	0.219	0.201

For evaluation, I will look at the WER and the *severity* (measured by the average cosine distance from Section) on the training and validation datasets throughout 50 epochs of training.

5.3.4 Results

Results show improvements in both severity (average cosine distance) and WER when incorporating severity into the loss function. The CTC-by-Cos Model, showed above an 85% improvement in severity and WER on the training dataset, and above an 18% improvement in severity and WER on the validation dataset (see Table 5.6). This improvement in performance suggests that there is potential to use severity in the development of ASR to decrease both the overall severity and WER.

CHAPTER 6

CONCLUSION

6.1 Discussion: Implications and Future Work

Because of the limitations in WER and based on the results of this work, I suggest looking at the average cosine distance between the embeddings ground-truth transcription and ASR output in conjunction with WER (Metric 3 in 5.3.3). However, when using this methodology one must be aware that the language models that generate text embeddings are **not** perfect, and can vary. Sentiment analyzers have shown to be very limited as they only capture one aspect or quality of an utterance, yet, these could still have their place in detecting errors in negation or certain words (i.e. “is” vs “isn’t” or “pain” vs “paid”).

In Experiment 2, WER generally agreed with the other metrics, but based on previous research done that shows the limitations of WER, one next area of research would be to study whether or not the metrics proposed in this work would be a better indicator of intelligibility or performance on an NLU task than WER [18, 54, 48].

This work also provides a theoretical basis for using text embeddings in the training of an ASR. Recent work has emerged showing that using semantic alignment (using text embeddings) for Spoken Language Understanding during training is very promising [28, 30]. Combining *severity* with the common loss function, Connectionist Temporal Classification, during the training regime of an ASR engine in Experiment

3 showed there is potential to optimize WER and severity at the same time for better results. With many machine learning methods, these results can vary greatly depending on the architecture, data, and training regime, therefore further experimentation is needed to more fully test the potential of this methodology.

I also see potential for text embeddings to be used more widely in evaluation beyond the field of ASR and into any field that involves the natural language generation such as in automatic text summarizing or even machine translation with the development of more multilingual language models [16]. Natural language generation is also important in healthcare in order to generate intelligible explanations for AI models [7], yet, similar to WER, in the field of natural language generation, common metrics such as ROGUE [33] and BLEU [42], have been shown to correlate poorly with human evaluations [7, 29]. Based on this work, I believe future work could involve incorporating these measures of severity in metrics for natural language generation tasks and studying how these correlate with human evaluations in a similar fashion to metrics like BERTScore or BLEURT [57, 50].

6.2 Summary

In this work I experiment with sentiment analyzers and text embeddings in the analysis of ASR errors. More specifically, I seek to answer the following questions: 1) do sentiment analyzers and/or text embeddings correlate with human ratings of severity, and if so, to what extent? 2) can we use sentiment analyzers and/or text embeddings as a useful measures or estimators of severity of ASR errors, 3) can we use these measures to evaluate the overall quality of the severity of an ASR engine?, and 4) can severity be useful in the development of an ASR system.

I first create my own data set of 150 ASR errors and 3 human ratings of severity by using an audio dataset of simulated patient-doctor conversations with transcriptions and 3 raters in the medical field. As a preliminary step, I show that there is decent consistency among raters.

To answer the first two questions, I use the difference in sentiment scores from 3 sentiment analyzers and the cosine distance of text embeddings from 4 language models as measures of severity. I look at the correlation between these measures and human ratings of severity as well as look at their ability to predict severity using a simple ordinal logistic regression. These are compared with WER, a common metric for evaluating ASR. While sentiment scores could not predict severity as well as WER, for all text embeddings models, the cosine distance from text embeddings to ASR output predicted severity better than WER.

I propose a simple method for incorporating these measures into metrics to evaluate the overall quality of an ASR engine. Generally the results agree with WER, (i.e. the ASR with the best WER performs the best on the metrics based on sentiment scores or cosine distance of text embeddings). I show that, upon deeper inspection, these metrics are capturing different qualities in ASR errors and can overcome some of the limitations of WER.

Finally, I experiment with incorporating severity into the development of an ASR system. Results show that there is potential for severity to help improve performance.

REFERENCES

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[2] Sadeen Alharbi, Muna Alrazgan, Alanoud Alrashed, Turkiayh Alnomasi, Raghad Almojel, Rimah Alharbi, Saja Alharbi, Sahar Alturki, Fatimah Alshehri, and Maha Almojil. Automatic speech recognition: Systematic literature review. *IEEE Access*, 9:131858–131876, 2021.

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016.

[4] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[5] Timo Baumann, Casey Kennington, Julian Hough, and David Schlangen. Recognising conversational speech: What an incremental asr should do for a dialogue system and how to get there. *Dialogues with Social Robots: Enablements, Analyses, and Evaluation*, pages 421–432, 2017.

[6] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[7] Alodie Boissonnet, Marzieh Saeidi, Vassilis Plachouras, and Andreas Vlachos. Explainable assessment of healthcare articles with QA. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[8] Herbert H Clark and Jean E Fox Tree. Using uh and um in spontaneous speaking. *Cognition*, 84(1):73–111, 2002.

[9] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Richard Dufour and Yannick Estève. Correcting asr outputs: specific solutions to specific errors in french. In *2008 IEEE Spoken Language Technology Workshop*, pages 213–216. IEEE, 2008.

[12] Rosa Falotico and Piero Quatto. Fleiss' kappa statistic without paradoxes. *Quality & Quantity*, 49(2):463–470, 2015.

[13] Faiha Fareez, Tishya Parikh, Christopher Wavell, Saba Shahab, Meghan Chevalier, Scott Good, Isabella De Blasi, Rafik Rhouma, Christopher McMahon, Jean-Paul Lam, et al. A dataset of simulated patient-physician medical interviews with a focus on respiratory cases. *Scientific Data*, 9(1):1–7, 2022.

[14] Shahla Farzana, Ashwin Deshpande, and Natalie Parde. How you say it matters: Measuring the impact of verbal disfluency tags on automated dementia detection. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 37–48, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[15] Shahla Farzana and Natalie Parde. Exploring mmse score prediction using verbal and non-verbal cues. In *INTERSPEECH*, pages 2207–2211, 2020.

[16] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[17] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[18] Olivier Galibert, Mohamed Ameur Ben Jannet, Juliette Kahn, and Sophie Rosset. Generating task-pertinent sorted error lists for speech recognition. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1883–1889, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[19] Sahar Ghannay, Yannick Esteve, and Nathalie Camelin. Word embeddings combination and neural networks for robustness in asr error detection. In

2015 23rd European Signal Processing Conference (EUSIPCO), pages 1671–1675. IEEE, 2015.

[20] Sahar Ghannay, Yannick Estève, and Nathalie Camelin. Task Specific Sentence Embeddings for ASR Error Detection. In *Interspeech 2018*, Hyderabad, India, September 2018. ISCA.

[21] Sahar Ghannay, Yannick Estève, and Nathalie Camelin. A study of continuous space word and sentence representations applied to asr error detection. *Speech Communication*, 120:31 – 41, 2020.

[22] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[23] David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W Black, Mosur Ravishankar, and Alexander I Rudnicky. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

[24] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.

[25] Keith Ito and Linda Johnson. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[26] Maree Johnson, Samuel Lapkin, Vanessa Long, Paula Sanchez, Hanna Suominen, Jim Basilakis, and Linda Dawson. A systematic review of speech recognition technology in health care. *BMC medical informatics and decision making*, 14(1):1–14, 2014.

[27] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[28] Sameer Khurana, Antoine Laurent, and James Glass. Samu-xlsr: Semantically-aligned multimodal utterance-level cross-lingual speech representation. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1493–1504, 2022.

[29] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 540–551, Hong Kong, China, November 2019. Association for Computational Linguistics.

[30] Gaëlle Laperrière, Valentin Pelloin, Mickaël Rouvier, Themos Stafylakis, and Yannick Estève. On the use of semantically-aligned speech representations for spoken language understanding. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 361–368, 2023.

[31] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[32] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[33] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[35] Karmele Lopez-de Ipiña, Unai Martinez-de Lizarduy, Pilar M Calvo, Blanca Beitia, Joseba Garcia-Melero, Miriam Ecay-Torres, Ainara Estanga, and Marcos Faundez-Zanuy. Analysis of disfluencies for automatic detection of mild cognitive impartment: a deep learning approach. In *2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI)*, pages 1–4. IEEE, 2017.

[36] Steven Loria et al. textblob documentation. *Release 0.15*, 2(8), 2018.

[37] Daniel Luzzati, Cyril Grouin, Ioana Vasilescu, Martine Adda-Decker, Eric Bilinski, Nathalie Camelin, Juliette Kahn, Carole Lailler, Lori Lamel, and Sophie Rosset. Human annotation of asr error regions: Is “gravity” a sharable concept for human annotators? In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 3050–3056, 2014.

[38] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[39] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[40] Kimberly D Mueller, Rebecca L Koscik, Bruce P Hermann, Sterling C Johnson, and Lyn S Turkstra. Declines in connected language are associated with very early mild cognitive impairment: Results from the wisconsin registry for alzheimer's prevention. *Frontiers in Aging Neuroscience*, 9:437, 2018.

[41] Hillary Ngai and Frank Rudzicz. Doctor XAvIer: Explainable diagnosis on physician-patient dialogues and XAI evaluation. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 337–344, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[43] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[44] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[45] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[46] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

[47] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[48] Giuseppe Riccardi and Allen L Gorin. Stochastic language models for speech recognition and understanding. In *ICSLP*, 1998.

[49] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[50] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

[51] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[53] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

[54] Ye-Yi Wang, Alex Acero, and Ciprian Chelba. Is word error rate a good indicator for spoken language understanding accuracy. In *2003 IEEE workshop on automatic speech recognition and understanding (IEEE Cat. No. 03EX721)*, pages 577–582. IEEE, 2003.

[55] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[56] Dong Yu and Li Deng. *Automatic speech recognition*, volume 1. Springer, 2016.

[57] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.