University of Sheffield

TITLE HERE

Adam Spencer

Supervisor: Professor JP Barker

A report submitted in fulfilment of the requirements
for the degree of BSc in Artificial Intelligence and Computer Science

in the

Department of Computer Science

May 6, 2023

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Adam Spencer

Signature:

Date: May 6, 2023

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Aims and Objectives

1.2 Overview of the Report

Chapter 2

Literature Survey

2.1 Automatic Speech Recognition

2.1.1 What is ASR?

Automatic Speech Recognition (ASR) is a technology which allows computers to recognise and produce a text transcription of spoken language. The research and development of technology involving speech has been a part of computer science since the late 1930s[1, 2], with rudimentary ASR systems being constructed as early as the 1950s[3]. These early attempts at recognising human speech treated it as a 'pattern matching' problem, the theory being that words could be constructed by matching the pattern created in a speech signal to corresponding spoken phonemes[1]. This paradigm falls apart when the system must be re-tuned for each individual, even for simple tasks such as recognising spoken digits[3], due to the fact that individual speakers don't produce exactly the same signal for each phoneme[4].

Since the 1970s, finding the solution to the problem of pattern matching for speech recognition has been considered unviable through the precise matching of patterns but instead finding the most probable pattern using statistical modelling[1]. The method which became most widely adopted and is still at the heart of modern ASR is Hidden Markovian Modelling, which was first applied to ASR in the '70s[5] and continued through the '90s[6] until today[7].

2.1.2 Hidden Markov Models

In his 1960 work[8], Dynkin describes a Markov process using the example of a randomly-moving particle in space;

" If the position of the particle is known at the instant t, supplementary information regarding the phenomena observed up till the instant t (and in particular, regarding the nature of the motion until t) has no effect on prognosis of the motion after the instant t (for a known "present", the "future" and the "past" are independent of eachother). "

From his description, we can draw the following assumptions for modelling a system as a Markov process;

- The system consists of states.
- The system is *in motion*, i.e. moving between states.
- This motion is random.
- The motion observed prior to t (say, $t-1$) does not influence $t+k$ where $k \geq 1$.
- Because the particle is constantly moving between states, the state at time t depends only on the state immediately prior, $t-1$.
- There is some probability, p, that the system moves from one state to another.

In a *Hidden* Markov Model (HMM), the states and transition probabilities between them are known, but for some ouput sequence the order and selection of states used to produce the output is not known. Knowing both the states and transition probabilities, it is therefore possible to calculate the most probable set of inputs used to produce the output.

To apply this model to speech, treat speech as a continuous sequence of discrete states, where each state is a feature vector representing an acoustic signal (either whole words, phonemes or even sub-phonetic features[6]). Assuming that each state is generated from a probabilistic distribution correlated with other states in the model[9] (i.e. probability that one state follows another) and having trained these distributions on known data, the output signal (i.e. the speech signal) can be used to determine the most probable sequence of tokens spoken. These tokens may then be decoded by a language model to construct a transcription[6].

Despite making up much of the research foundational to modern ASR, Markov models have a crucial flaw when applied to speech; parts of speech are dependent on more than just the part immediately before (i.e. $t-1$). For instance in a presentation discussing *hats* it is unlikely that the word *cat* would be used, despite the phonetics of the word being largely the same.

2.1.3 How Does Modern ASR Work?

Skipping ahead from the mid-1980s to the current day, ASR has moved towards what is known as the 'end-to-end' or 'encoder-decoder' model; at a high level, the input audio is encoded into features, these features are aligned with language and then decoded to produce an output transcript[10]. The key difference between modern approaches and the classical HMM-based approach is the use of widely researched 'machine-learning' techniques, including various forms of neural network[11, 12, 13, 14].

Recent research has proposed a new network architecture called the *Transformer*[15], aiming to reduce the computational complexity of encoder-decoder models by forgoing convolutional or recurrent neural networks (CNNs and RNNs) and instead relying on 'self-attention'. The motivation for the *Transformer* can be understood as follows;

- CNNs (e.g. [16]) and RNNs (e.g. [17]), while popular, have greater per-layer computational complexity than self-attention[15].
- Recurrent neural networks must perform $O(n)$ sequential operations for a sequence length n, whereas self-attention has a constant (i.e. $O(1)$) maximum number of sequential operations, enabling parallel computation[15].
- By allowing each layer in the encoder and decoder to attend to the whole output of the previous layer,

In a multilayer network, self-attention layers are used to build relations between seperate parts of an input sequence by allowing each node (or 'attention head'[18]) to attend to all outputs from the previous layer.

According to the work that introduced it[15], attention in the Transformer architecture is calculated as;

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

Where;

- Q is the Query
- ...

2.1.4 Problems in ASR

Despite their ubiquity, modern ASR systems aren't without fault. Cutting edge systems like *(*wav2vec) are touted as being capable of achieving 'greater-than-human' scores on specific datasets[19, 20, 21] such as *LibriSpeech*[22], achieving as low as 1.4% error[23].

A major problem with comes when the data is not 'clean', for example, background noise is present, microphones are far away, the speaker has an atypical speech pattern, etc. In this setting, *wav2vec* achieves much poorer scores with word error rates as high as 65%[24] on the *CHiME6* corpus[25].

2.1.5 Whisper

In late September 2022, the OpenAI research laboratory (known for such projects as GPT-3/4 and ChatGPT) released a new open-source ASR system known as 'Whisper' [24]. Whisper is unique in being very large (trained on 680,000 hours of speech data), open-source, and fully supervised; all the training data used to create the model has been accurately labeled and quality-checked by humans, unlike much larger unsupervised (or semi-supervised) models such as 'BigSSL' (1,000,000+ hours of data) [20].

Unsupervised training is appealing for training speech recognisers because there is a wealth of unlabeled recordings, and labeled recordings are uncommon for less widely-spoken languages[26]. Unsupervised systems have a clear disadvantage, however, when compared to supervised; they lack clear decoder mappings[24], meaning that even for a succesfully encoded input there may not be a clear mapping from that input into a speech token. To solve this, fine-tuning to map encodings to decodings is done on the part of the model's developers, though this is a precarious route to overfitting; if the model is too fine-tuned to its training data, performance will suffer when faced with data which isn't well represented in the training set.

For example, if an unsupervised model were trained using the voices of young people, it may perform with considerably poorer accuracy when used to transcribe elderly speakers due to differences inherent to their speech[4].

Whisper uses a natural language model to perform next-token prediction (in layperson's terms, there is a secondary system trying to ensure the intelligibility of sentences produced from transcription). In a practical setting this means that conversational speech (i.e. speech which flows as sentences rather than semantically-disjoint terms) should be transcribed with a higher degree of accuracy.

2.2 ASR Confidence

2.3 Speech Corpora

2.4 Understanding Transcription

2.4.1 Manual Transcription

2.4.2 Semi-Automatic Transcription

2.4.3 Fully-Automatic Transcription

2.5 Summary

Chapter 3

Requirements and Analysis

3.1 Project Requirements

The objective of this work is not to present a new paradigm through which transcription may develop, nor is it to produce a fully-working, infallible system which aims to receive actual use by transcribers. Rather, the aim of this work is to explore the current state of the field of ASR and to understand the extent that current ASR technology could provide aid to a human transcriber.

To further understand its objective, the requirements of this work are as presented in the following subsections.

3.1.1 Motivate computer-aided transcription

Before exploring how a computer system may aid a human transcriber, it is important to understand;

- *why* a human transcriber may require aid;
- to *whom* a computer-aided transcription system would provide benefit; and
- what the *extent* of such a benefit would be.

3.1.2 Generate transcripts using ASR

Evaluating the quality of ASR transcription requires a key set of data; ASR-generated transcripts. Rather than comparing different ASR systems, Whisper[24] has been chosen as the only system to use for generating transcripts because;

- it is new (made available in September 2022);
- it is entirely free and open-source, meaning it is easily modifiable and available to be used without licence; and

- it reportedly achieves very good results across different speech corpora.

Whisper[24] is implemented in Python using the PyTorch[27] library which allows computation to take place on GPUs which support CUDA, meaning transcripts can be generated very quickly on a high-performance computer (HPC) system. Luckily, the University of Sheffield offers access to HPC clusters[28], and I have been granted access to the *Bessemer* cluster for the completion of this project.

The key to generating useful transcripts is some high-quality speech recordings from a speech corpus. While preliminary testing of Whisper may use data from any available corpora, it would be very useful to obtain some data which is;

- not present in Whisper's training data, as to prevent the model from regurgitating labels for data it has already seen; and
- is well suited to the task of computer-aided transcription, as to enable more practical evaluation.

It is also useful, however, to understand the caveats related to using well-suited data! The naïve assumption that all data seen by *any* computer system is 'perfect' would misrepresent the usefulness of the system in question. To combat this, this work must acknowledge the limited extent to which a computer-aided transcription system using Whisper is viable, and evaluate how the viability could be increased to be more applicable to real-world tasks.

3.1.3 Experiment with confidence measures

Neural network confidence is widely discussed in the literature. Rather than attempt to create a novel method for calculating a system's confidence, this work should focus on re-creating and applying existing measures of confidence to Whisper, whether through modification to the model itself or through inference of the model's output.

It would be of great utility to understand how system confidence may aid a human transcriber, as well as how exactly 'confidence' shall be defined in the case of this specific task. Such understandings would further refine the 'lens' through which the system may be evaluated, and as such are vital to the completion of this work.

3.1.4 Explore designs for computer-aided transcription

There's limited use in a purely theoretical exploration of a computer system which is designed to be interfaced with by a human, such as this. Providing various design concepts for a real computer system shall aid the reader in grasping the benefit of a

3.2 Analysis

This section shall provide analyses of the requirements cited in 3.1, presenting an analysis of each aforementioned requirement in corresponding order to that in which the requirements are given. The general intention of this section is to determine the lens through which the project shall be evaluated, and to contribute a detailed motivation for the decisions made in both the design and implementation of the project.

The following subsections constitue an analysis of the aims of the project.

3.2.1

3.2.2

3.2.3

3.2.4

3.3 Ethical, Professional and Legal Issues

Chapter 4

Design

4.1 Risk Analysis

4.2 Project Plan

Chapter 5

Implementation and Testing

5.1 Preparing the Data

While the LifeLUCID corpus[29] consists of conversational audio recordings, each of these recordings are presented as individual stereo WAVE files approximately 10 minutes in length, with each speaker recorded seperately in either the left or right channel. Time-aligned transcriptions accompany these data in *Praat TextGrid* format.

5.1.1 The TextGrid Format

Praat is a piece of software for speech recording and analysis[30] and a *TextGrid* is used to align individual *speech tokens* with the time in which the are uttered in the recording. When viewed in a text editor, *TextGrid* files appear as a descending series of intervals, indexed in the order they occur; with start- and end-times, and individual speech tokens. To illustrate the format, here is a snippet taken from *LifeLUCID*, the utterance is simply "a bush with a yello duck on top";

```
intervals [12]:
  xmin = 20.899
  xmax = 20.971783458461772
  text = "SIL"
intervals [13]:
  xmin = 20.971783458461772
  xmax = 21.05
  text = "a"
intervals [14]:
  xmin = 21.05
  xmax = 21.47
  text = "BUSH"
intervals [15]:
  xmin = 21.47
```

```
  xmax = 21.66
  text = "with"
intervals [16]:
  xmin = 21.66
  xmax = 21.720024609817834
  text = "A"
intervals [17]:
  xmin = 21.720024609817834
  xmax = 22.1
  text = "SIL"
intervals [18]:
  xmin = 22.1
  xmax = 22.49
  text = "yellow"
intervals [19]:
  xmin = 22.49
  xmax = 22.84
  text = "duck"
intervals [20]:
  xmin = 22.84
  xmax = 23.06
  text = "ON"
intervals [21]:
  xmin = 23.06
  xmax = 23.769
  text = "top"
```

Considering that this file contains over 1000 of these intervals, this example should hopefully demonstrate that the *TextGrid* format is not particularly readable. In order to simplify quality checking as well as to allow more accompanying metadata (e.g. ASR results), the utterances shall be moved into *JSON* format.

Due to Whisper being written entirely in Python, to maintain language-homogeneity a Python library named `textgrid.py`[31] was used to read and manipulate TextGrid files rather than dealing with the transcription data using *Praat*.

According to their documentation, the *TextGrid* files for LifeLUCID[29] contain some special, non-speech tokens to denote certain parts of the speech recordings as follows:

- `<SILP>` denotes time where one participant is silent and the other is talking.
- `<SIL>` denotes silent time between words, where the speaker is silent but the other participant is also silent, such as when the speaker is taking a breath.
- `<GA>` denotes either the time before the task begun but the recording had started or external noises picked up by the microphone.

- `<BELL>` replaces moments when a participant has pressed their bell, these moments are also silent in the recording.

Given that these special tokens are marked by the times at which they begin and end, it was possible to segment the large audio files into hundereds of short utterances.

5.1.2 Generating Utterances

The contents, beginning, and end of every utterance where computed using the data available in the *TextGrid* files using a Python script named `get_utterances`. This script operates over a directory containing *TextGrid* files, writing out the utterances as files in *JSON* format.

The script also takes as args; a minimum time between tokens required to end the utterance and a maximum pause time allowed within one utterance. These thresholds allow utterances to be fine-tuned by a user, leading to fewer drawn-out or unreasonably short utterances.

JSON was selected due to its ability to be easily read and understood by a human, unlike *TextGrids*. This allowed for simple verification of the data without the need for more specific software to view the files.

5.1.3 Audio Segmentation

Another Python script named `segment_audio` was created to generate audio files for each utterance. Given two directories as input; one containing `.json` files (as output by the `get_utterances` script) and the other containing `.wav` files representing each audio recording, the audio is split along the beginning and end times of each utterance and output to a new directory.

This script uses the *python-soundfile* module[32] to load audio files into *NumPy*[33] arrays. By multiplying the sampling rate of the audio by the start- and end-times of each utterance, the array indices at the start and end of each utterance are computed. Array slices between these indices represent each utterance, which can then be saved to new audio files using the *python-soundfile* module.

5.2 ASR With Whisper

Whisper is available as a Python module named `whisper`[34]. The module features a `transcribe()` function to transcribe audio files given as a parameter to the function and return an object containing the output of Whisper.

Chapter 6

Results and Discussion

Chapter 7

Conclusions

Bibliography

[1] L. R. Rabiner, "Automatic Speech Recognition - A Brief History of the Technology Development," *Scinapse*, Jan. 2004.

[2] D. W. H., "The vocoder," *Bell. Labs. Rec.*, vol. 18, p. 122, 1939.

[3] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *The Journal of the Acoustical Society of America*, vol. 24, no. 6, pp. 637–642, 1952.

[4] W. S. Horton, D. H. Spieler, and E. Shriberg, "A corpus analysis of patterns of age-related change in conversational speech.," *Psychol. Aging*, vol. 25, no. 3, p. 708, 2010.

[5] J. K. Baker, *Stochastic modeling as a means of automatic speech recognition.* Carnegie Mellon University, 1975.

[6] Y. Bengio *et al.*, "Markovian models for sequential data," *Neural computing surveys*, vol. 2, no. 199, pp. 129–162, 1999.

[7] X. Dong, W. Cao, H. Cheng, and T. Zhang, "Hidden markov model-driven speech recognition for power dispatch," in *Tenth International Conference on Applications and Techniques in Cyber Intelligence (ICATCI 2022)* (J. H. Abawajy, Z. Xu, M. Atiquzzaman, and X. Zhang, eds.), (Cham), pp. 760–768, Springer International Publishing, 2023.

[8] E. B. Dynkin, *Theory of Markov Processes.* Pergamon Press, 1960.

[9] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[10] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, p. 1018, 2019.

[11] M. K. Mustafa, T. Allen, and K. Appiah, "A comparative review of dynamic neural networks and hidden markov model methods for mobile on-device speech recognition," *Neural Computing and Applications*, vol. 31, pp. 891–899, 2019.

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

[13] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," *arXiv preprint arXiv:1706.02737*, 2017.

[14] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4835–4839, IEEE, Mar. 2017.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[16] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *arXiv preprint arXiv:1812.06864*, 2018.

[17] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, pp. 1764–1772, PMLR, 2014.

[18] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[19] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 12449–12460, Curran Associates, Inc., 2020.

[20] Y. Zhang, D. S. Park, W. Han, J. Qin, A. Gulati, J. Shor, A. Jansen, Y. Xu, Y. Huang, S. Wang, Z. Zhou, B. Li, M. Ma, W. Chan, J. Yu, Y. Wang, L. Cao, K. C. Sim, B. Ramabhadran, T. N. Sainath, F. Beaufays, Z. Chen, Q. V. Le, C.-C. Chiu, R. Pang, and Y. Wu, "BigSSL: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1519–1532, oct 2022.

[21] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," 2021.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "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, 2015.

[23] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, “Pushing the limits of semi-supervised learning for automatic speech recognition,” 2020.

[24] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision.” arXiv:2212.04356, 2022.

[25] S. Watanabe, M. I. Mandel, J. Barker, and E. Vincent, “Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings,” *CoRR*, vol. abs/2004.09249, 2020.

[26] A. Baevski, W.-N. Hsu, A. CONNEAU, and M. Auli, “Unsupervised speech recognition,” in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 27826–27839, Curran Associates, Inc., 2021.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[28] “High Performance Computing at Sheffield — Sheffield HPC Documentation,” May 2023. [Online; accessed 6. May 2023].

[29] O. Tuomainen, L. Taschenberger, and V. Hazan, “LifeLUCID Corpus: Recordings of Speakers Aged 8 to 85 Years Engaged in Interactive Task in the Presence of Energetic and Informational Masking, 2017-2020,” *UK Data Service*, May 2021.

[30] “Praat: doing Phonetics by Computer,” Mar. 2023. [Online; accessed 8. Apr. 2023].

[31] K. Gorman, “textgrid.py,” Apr. 2023. [Online; accessed 8. Apr. 2023].

[32] B. Bechtold, “python-soundfile,” 2013. [Online; accessed 8. Apr. 2023].

[33] C. R. Harris, K. J. Millman, S. f. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020.

[34] “openai-whisper,” Apr. 2023. [Online; accessed 10. Apr. 2023].

Appendices

Appendix A

An Appendix of Some Kind

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc.

Appendix B

Another Appendix

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc.