

Automated Detection of Personally Identifiable Information on Speech Data

Taras Andrushko
Supervisor: Pierre Lison



Thesis submitted for the degree of
Master in Language Technology
60 credits

Department of Informatics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Autumn 2025

Automated Detection of Personally Identifiable Information on Speech Data

Taras Andrushko
Supervisor: Pierre Lison

© 2025 Taras Andrushko
Supervisor: Pierre Lison

Automated Detection of Personally Identifiable Information on Speech
Data

<http://www.duo.uio.no/>

Printed: Reprocentralen, University of Oslo

Declaration of the use of Generative Artificial Intelligence

In this research paper, Generative Artificial Intelligence has been used. All interactions with Generative Artificial Intelligence were carried out in accordance with The University of Oslo's regulations. I as an author of this paper take full responsibility for the content, references and claims in this document.

- Improving paper readability by correcting formulation in some parts of the work.
- Generating tables and plots.
- Generating and debugging parts of the code.

All of the content, text, and code generated with Gen. AI was checked, reviewed and approved before usage.

Abstract

Automatic detection of personally identifiable information (PII) in speech is an important step toward secure and privacy-aware conversational AI. This thesis investigates the task of identifier detection using both binary and multilabel classification approaches based on speech representations.

The study compares pretrained speech models — WavLM, HuBERT, and Whisper — combined with different pooling strategies, including hierarchical, gated and multihead attention pooling, trained with different input sizes. Evaluation was performed both on Silver automatically annotated spoken datasets and manually annotated Gold Standard. Results show that the Whisper model with multihead attention pooling achieved the best performance with wider input sizes, both in multilabel and binary setups. For narrower input sizes, the Hubert and WavLM models appeared to be a better solution than the Whisper one. Results also allowed us to determine the value of context for different setup settings, both within binary and multilabel approaches. Comparison with binary classifiers demonstrated that multilabel setups can also provide granularity while maintaining competitive detection accuracy.

The work highlights the benefits of attention-based pooling for PII classification with only audio feature input task and outlines future directions for improving model generalisation and class balance.

All source code for the best multilabel and binary approaches, together with results and the collected datasets (Silver + Gold Standards), is publicly available in a dedicated GitHub repository¹.

¹https://github.com/letitself/Speech_DI_Audiofeatures_only

Contents

1	Introduction	1
1.1	Research questions and hypotheses	2
1.2	Contributions	3
1.3	Overview	4
2	Literature Review	5
2.1	Background	5
2.2	Task definition and terminology in data anonymisation	7
2.3	Existing approaches for text data	9
2.4	Existing approaches for speech	13
2.5	Gaps in the Literature	16
3	Motivation and Approach	17
3.1	Motivation and Challenges	17
3.2	Task formulation and possible solution	18
3.3	Datasets	19
3.4	Approach	19
3.5	Evaluation	21
4	Setup and Methodology	23

4.1	Data selection	23
4.1.1	Data crawling	25
4.1.2	Data pre-processing	26
4.2	Silver Standard, Dataset Structure and Insights	28
4.3	Gold Standard	31
4.4	Setup	35
4.4.1	Baseline	35
4.4.2	Input window	36
4.4.3	Context	41
4.4.4	Binary classification setup and data distribution . . .	42
4.4.5	Multilabel classification setup and data distribution .	43
4.5	Optimisation, hyperparameters and Pooling Techniques . .	45
4.5.1	Pooling Techniques	46
4.6	Evaluation Metrics	49
4.7	Models	51
4.7.1	WavLM model	52
4.7.2	Hubert model	55
4.7.3	Whisper model	57
5	Results	59
5.1	Multilabel results	59
5.1.1	Multilabel Phrase input window	60
5.1.2	Multilabel Word input window	61
5.1.3	Multilabel Equal input window	63
5.1.4	Multilabel Frame input window	64

5.1.5	Summary Multilabel Results	66
5.2	Binary results	68
5.2.1	Binary Phrase input window	68
5.2.2	Binary Word input window	70
5.2.3	Binary Equal input window	72
5.2.4	Binary Frame input window	73
5.3	Summary for Binary results	75
5.4	Result Summary	77
5.5	Full dataset training Results	79
5.6	Base-line vs. Presidio vs. Custom approaches	79
5.7	Discussions	82
5.7.1	Pooling techniques influence	82
5.7.2	Input size trade-offs for models	83
5.7.3	Context influence on Model's performance	85
6	Conclusion	86
6.1	Limitations	87
6.2	Future directions	89

List of Figures

3.1	de-ID Pipeline from (Cohn et al., 2019a)	20
3.2	Figure of Approach	20
4.1	Distribution Figure	26
4.2	Pipeline for dataset preparation with Whisper and Presidio .	27
4.3	Distribution Figure	34
4.4	Phrase and Word Level Slicing	36
4.5	Frame and Equal Slicing	38
4.6	Model input	41
4.7	Distribution Figure	42
4.8	Filtered Distribution	44
4.9	Training pipeline schema.	45
4.10	Figure from WavLM paper (Chen et al., 2022)	53
4.11	Figure from original Hubert paper (Hsu et al., 2021)	56
4.12	Figure from original Whisper paper (Radford et al., 2022) . .	57

List of Tables

2.1	Technical Comparison of Voice Anonymisation Methods . . .	14
4.1	Dataset summary: Full vs Demo	29
4.2	General and USA-Specific PII Types	33
4.3	Comparison of input sizes for PII detection models	40
4.4	Compact summary of evaluation metrics	50
5.1	Phrase-level multilabel PII classification results for different models and pooling strategies.	60
5.2	Word-level multilabel PII classification results for different models and pooling strategies.	62
5.3	Equal-slicing (0.5 s) multilabel PII classification results for different models and pooling strategies.	63
5.4	Frame-wise multilabel PII classification results for different models and pooling strategies.	65
5.5	Binary phrase-level PII classification results for different models and pooling strategies.	69
5.6	Binary word-level PII classification results for different models and pooling strategies.	71
5.7	Binary equal-slicing (0.5 s) PII classification results for different models and pooling strategies.	72
5.8	Binary frame-wise PII classification results for different models and pooling strategies.	74

5.9	Summary of best-performing approaches for each input size in binary and multilabel setups.	77
5.10	Performance summary of the model.	79
5.11	Comparison of Multilabel and Binary models on unseen data.	80
5.12	Comparison of Presidio, Binary, and Multilabel models against the Gold Standard.	81
1	Comprehensive summary of binary and multilabel PII classification results across models, pooling strategies, context, and granularity. For multilabel, we report mAP, AUC (macro/micro), and F1 (macro/micro). For binary, we report Precision, Recall, and F1.	96

Chapter 1

Introduction

The task of Data Anonymisation and its key importance in the world of developing technologies cannot be overstated. In today's digitalised society, both public and private organisations rely on a vast collection of different kinds of data. Organisations can protect sensitive data with different anonymisation techniques, which could prevent privacy breaches, identity theft and financial fraud. Several legal frameworks, such as General Data Protection Regulation (GDPR) (Voigt & Von dem Bussche, 2017) in the European Union, and ('Health Insurance Portability and Accountability Act of 1996', 1996) in the United States, mandate strict control on PII collection, its processing and storage within different sectors (health, finance, and education).

The rapid rise of Generative AI platforms - such as ChatGPT¹, DeepSeek², and Gemini³ - has transformed how data is processed and stored, but it has also introduced new challenges for safeguarding sensitive information. Authors of (Cheng et al., 2021) mention that Large Language Models (LLMs) are trained on massive and diverse datasets, raising concerns that individuals could be inadvertently exposed through the data used to train them. In the midst of rapidly advancing generative AI, anonymisation systems serve as an important safeguard for protecting personal data and also a tool that can potentially allow the use of training data that was previously prohibited by legal regulations.

The task of anonymisation is complex and multimodal, requiring consideration of many factors, such as what is considered as Personal Identifier Information (PII) and what does not lead to personal identification. The

¹<https://chatgpt.com/>

²<https://www.deepseek.com/en>

³<https://gemini.google.com/app>

challenge of anonymisation is a trade-off between preserving contextual information while masking critical elements that could, when combined, lead to identity disclosure. In this paper, we will specifically discuss methods of anonymising audio and text data.

The task of anonymisation in the text domain is directly related to the classification of different types of PII (names, locations, numbers etc.). The problem of anonymising text data is widespread and has many different approaches to its solution, both rule-based and machine learning-based, which are also described in this work in Chapter 2. When anonymising text, the main task is to classify potential PII (this could be a word, name, or number that could lead to the disclosure of an individual) and remove it or replace it with an alternative.

Unlike text, audio anonymisation operates in the audio data domain and pursues a different goal. In the Chapter 2, in addition to an overview of approaches to text anonymisation, we also analyse existing audio approaches, where we note that the task of audio anonymisation is primarily an attempt to disguise audio characteristics of the speaker and mainly focuses on the task of voice characteristics transformation.

Although altering the speaker's voice characteristics is an important anonymisation task, it leaves a gap by not working with the content the speaker utters. Therefore, even a modified speaker's voice is not a solution that prevents identity disclosure, since potentially undeleted PII in the audio itself can still reveal the individual. This particular gap in research, which will be discussed in detail in Section 3.1, shapes the direction of this work and raises several research questions and hypotheses which we will try to answer with our proposed approach.

1.1 Research questions and hypotheses

In this work, we will attempt to develop an approach to classify PII in the audio segment. The approach will be aimed at finding sensitive information in the text using fine-tuned encoder-based models with only audio input.

The main goal of this work is to test the hypothesis that encoder-based models can classify PII using only audio input. In other words, we aim to design a model that processes audio data in a similar way to how text-based approaches handle textual input - receiving an audio segment as input and outputting a label indicating whether it contains PII or not.

We hypothesise that modern encoder-based models - Whisper⁴, WavLM⁵, and HuBERT⁶ - can classify audio inputs as PII or non-PII in both multilabel and binary classification settings. We also think that some of the models that we have chosen for this research will perform differently depending on the input size, and some of the models will classify better for shorter inputs. An additional context to the input, in our opinion, can be a crucial factor which can affect performance results. To explore this hypothesis and evaluate the potential of encoder-based models for audio-based PII classification, the following research questions are proposed:

1. **RQ1:** How accurately can models identify the presence of PII in speech based only on audio features input?
2. **RQ2:** How do different encoder-based models (Whisper, WavLM, and HuBERT) perform with different audio input sizes?
3. **RQ3:** How can additional context to the inputs affect model performance?
4. **RQ4:** To what extent can audio-based PII detection operate as a multilabel classification problem (distinguish between different types of PII)?

These four research questions will form the frame of this work and help us to understand whether it is possible to classify audio inputs as PII or No-PII with encoder-based models.

1.2 Contributions

The main thesis contributions are as follows:

1. An overview of existing text and speech anonymisation approaches (Chapter 2).
2. Collection, annotation and labelling of a new corpus for the task of automatic PII detection in speech data task using only audio input. (Available on the dedicated GitHub Repository⁷)

⁴<https://huggingface.co/openai/whisper-base>

⁵<https://huggingface.co/microsoft/wavlm-base-plus>

⁶<https://huggingface.co/facebook/hubert-base-ls960>

⁷Corpus for PII classification with audio features only - https://github.com/letitself/Speech_DI_Audiofeatures_only

3. Developing an approach for automatic PII detection (Chapter 4).
4. Evaluation and analysis of 3 encoder-based models' performance within the task of automatic PII detection in speech data trained on the collected and annotated corpus (Chapter 5).
5. Fine-tuned Automatic PII detection models both for binary and multilabel tasks.(Available on the dedicated GitHub Repository ⁸)

1.3 Overview

The thesis has the following structure:

- **Chapter 2:** Literature review.

The chapter provides an overview of existing research and current approaches within the task of anonymisation, both for text and audio domains.

- **Chapter 3:** Motivation and Approach.

The motivation chapter explains why we need better, more flexible PII detection systems for real-world use, outlining the importance of this work.

- **Chapter 4:** Setup and Methodology.

This chapter outlines the experimental design, datasets, model architectures, and evaluation metrics used in the research.

- **Chapter 5:** Results.

This chapter provides a discussion of quantitative and qualitative outcomes from the conducted experiments, as well as a comparative analysis of the three models in different training configurations.

- **Chapter 6:** Conclusion.

This chapter summarizes the main insights of this work, addressing the limitations, and offering recommendations and directions for future research.

⁸Finetuned Models - https://github.com/letitself/Speech_DI_Audiofeatures_only

Chapter 2

Literature Review

2.1 Background

The development of computer technology has greatly simplified a person's life, but at the same time, huge amounts of information have begun to be collected. Such information can greatly threaten a person's privacy because it may contain a person's data, including their name, surname, phone number, and so on.

Data is a necessary resource for the task of training machine learning models. However, it often remains unclear what specific data is used to train a particular algorithm. The developers of one of the leading LLMs, OpenAI¹, do not disclose what data they use to train their model – ChatGPT, and we cannot be completely sure that the model was not trained on our personal data.

Various sources can contain not only personal data information (name, date of birth, etc.), but also personally identifiable information (medical, educational, financial, etc.). For example, if you have any popular social media account such as Facebook, WhatsApp, or Instagram, you share not only your web-browsing information and purchase history with the media-giant Meta, but also your personal information such as name, telephone number, e-mail address and actual address. According to Meta's Privacy Policies ² they collect this information only for personalising ads and suggested content. According to those Policies, they also utilise user data/content for training their recent generative AI models.

¹<https://openai.com/>

²Meta's Privacy Policies - <https://www.facebook.com/privacy/policy/>

Another source of sharing your personal information is through research interviews conducted by scientists to study a particular issue. For example, interviews or questionnaires that are compiled by social linguistics researchers, usually contain questions regarding place of residence, level of education, etc. (Mallinson et al., 2017).

In general, in today's world, we often share our personal information during our daily routine. Opening a new bank account, going to the dentist, getting insurance, buying a plane ticket, paying utilities, paying taxes, registering an account in a web service, everywhere you share at least your phone number or email address, and often your name, age, and even biometrics.

The General Data Protection Regulation (GDPR) (Voigt & Von dem Bussche, 2017) is a comprehensive data protection and privacy law implemented by the European Union in 2018. This law is vital to protecting people's personal data and ensuring that the data itself is collected and stored correctly and securely. But why is it so important to protect personal data and regulate how it is collected and how much the person from whom it is shared knows about it?

The main threat behind personal data disclosure is the potential for privacy violations and misuse of sensitive information. Misuse of sensitive information can have social, financial, and psychological consequences for the affected individuals; therefore, anonymisation can prevent those potential threats. However, even anonymised data can still be a source and help in the person re-identification task. Article 29 DATA PROTECTION WORKING PARTY (Graux & Graux, 2018) highlights the importance of reducing re-identification risks and protecting personal data.

Anonymisation alone does not eliminate all threats, and re-identification is one of them. As also stated in ARTICLE 29 Data Protection Working Party, "...even anonymised data, like statistics, may be used to enrich existing profiles of individuals, thus creating new data protection issues...". This demonstrates the importance of the proper setup of an anonymisation system, which could act as a preventative measure and can handle potential re-identification risk in the future.

Beyond re-identification, the disclosure of personal data can also result in discrimination and reputational damage. Individuals may face unfair treatment by unintended exposure of their data (Graux & Graux, 2018). These potential harms reinforce the need for proactive privacy-preserving technologies — such as automated detection of personally identifiable information (PII) - that can prevent sensitive content from being disclosed before anonymisation even takes place.

The development of effective anonymisation techniques is crucial not only for safeguarding personal information but also for enabling the responsible use of data across various domains. By removing or obscuring identifiers, anonymisation reduces privacy risks, making it possible to share, analyse, and extract insights from data without compromising individual identities. Those benefits are important within fields such as healthcare, finance, and social sciences, where access to rich datasets enables advancements in research, policy-making, and innovation, while complying with stringent privacy regulations like the GDPR and HIPAA.

After we have established the importance of the data anonymisation problem, we can now examine the key concepts and definitions of the anonymisation process.

2.2 Task definition and terminology in data anonymisation

First of all, it is worth defining what personal data and identifiers are. Personal data is any data that can tell something about a person. Such data can contain several attributes/types of identifiers which are mentioned in (Domingo-Ferrer et al., 2022):

- 1. Direct identifiers:** a type of identifier that allows you to identify an individual without additional information. Such identifiers are usually very explicit in relation to a specific person. Examples of such identifiers are: social security number, passport number, etc.
- 2. Indirect identifiers:** also known as quasi or pseudo identifiers, these are pieces of information that on their own may not indicate a specific person, but in combination with other data can be used to reveal the person's identity. This type of attribute is complicated by the fact that essentially everything that is indirectly connected with a person can be called a quasi-identifier and lead to a re-identification of the person. Examples of quasi-identifiers are postal code, gender, and date of birth.
- 3. Confidential attributes:** sensitive information on the individuals that took part in the data collection process is called confidential attributes. Examples of confidential attributes could be information regarding the salary of the person, health condition, and sexual orientation.
- 4. Non-confidential attributes:** everything else that does not belong to any type of the attribute described above is considered to be non-

confidential. Basically, this type of identifier does not lead to a person's disclosure.

Notably, some of the categories are not mutually exclusive and indirect identifiers could also be confidential or non-confidential attributes at the same time.

It is also worth mentioning that a particular writing/talking style could be used as an identifier, and it also needs to be considered as a serious threat. However, in this work, we will focus only on the categories mentioned above.

In (Lison et al., 2021b) the authors distinguish three closely related but legally and technically distinct concepts in data privacy:

1. Anonymisation: According to GDPR, it is a process of complete and irreversible removal or alteration of personal identifiers and any other information from a data source that could lead to identification either directly or indirectly. The process of anonymisation must be irreversible and potentially resistant to re-identification of the individual by combining the anonymised data with other external datasets. However, achieving true anonymisation is a very challenging task, especially for high-dimensional or linked datasets (where risks remain even after applying anonymisation methods to it) (Ohm, 2009).

2. De-identification: In this type of anonymisation, identifiers are removed or masked only within a predefined set of direct identifiers (names, cities, etc.); meanwhile, indirect or quasi-identifiers often remain in de-identified data. However, de-identified data does not guarantee anonymity. De-identification techniques are usually used in the healthcare data sharing domain, where regulatory laws, such as the aforementioned HIPAA, define standards for particular identifiers which should be removed or masked. (El Emam, 2013). Unlike anonymisation, de-identified data remains an object of personal data; therefore, it is still protected by GDPR regulations.

3. Pseudonymisation: this particular technique is intended to replace identifiers with pseudonyms or coded values, which can be reversed in the future. Unlike anonymisation, pseudoanonymised data, as well as identified one, also remains an object of personal data and protected by GDPR regulations. Pseudonymisation allows safer data processing, but it is not equivalent to anonymisation. This particular technique requires additional safeguards to prevent unintended and unauthorised re-identification (Voigt & Von dem Bussche, 2017).

Thus, the choice among these data transformation techniques depends

on the intended use, balancing data utility with privacy protection and regulatory compliance. Research within the field of anonymisation highlights that imperfect anonymisation or weak pseudonymisation could lead to re-identification and person disclosure threats (Lison et al., 2021b), (Ohm, 2009).

Knowing the basic concepts and definitions of the anonymisation task, we can consider what modern approaches exist and what challenges introduce data inputs.

2.3 Existing approaches for text data

There are plenty of anonymisation techniques used in the task of data anonymisation for text data, each having its own advantages and disadvantages.

In the recent papers (Lison et al., 2021a), (Asimopoulos et al., 2024), (Deußer et al., 2025a) authors describe different anonymisation approaches ranging from hand-crafted methods to more recent and advanced deep-learning models, highlighting their advantages and disadvantages. In this research work, we will provide a short overview of the approaches mentioned in those papers in the following paragraphs.

Rule-based and dictionary-based techniques. Manually crafting rules and compiling dictionaries or gazetteers of sensitive terms, phrases, patterns, etc. that should be detected and masked or replaced during anonymisation. These methods are very straightforward and easy to interpret. The main disadvantage of this approach is the lack of scalability and adaptation to new or ambiguous data types.

Conditional Random Fields (CRF). In the context of text anonymisation, CRFs can be trained to detect and label sensitive entities or tokens that require anonymisation.(Raj & D’Souza, 2021). It is widely used in identifying sensitive entities such as names, dates, or locations. CRFs are flexible with feature selection, allowing the integration of linguistic and orthographic cues that improve detection accuracy (Leevy et al., 2020). A disadvantage of this approach is that it heavily depends on the design of features, struggles with unknown words, is biased towards frequent labels and requires a lot of computational power during the training process. Despite those disadvantages CRF approach remains a strong approach as a baseline within the task of text anonymisation.

Named Entity Recognition (NER) models. The aim of NER models is

to identify and classify named entities like person names, locations, etc., which can be useful for anonymising direct identifiers. This approach is more adaptive to different datatypes and has the ability to adjust dynamically to varying contexts(Leevy et al., 2020). In this benchmark (Asimopoulos et al., 2024), an NER model was used as a baseline for machine learning approaches.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of capturing long-range dependencies in sequential data like text. In Benchmark (Asimopoulos et al., 2024) LSTM approach was used as a baseline for the traditional neural approach for the anonymisation task. This approach is suitable for an anonymisation task since it captures the context of a potential identifier. In comparison to traditional approaches, LSTM provides better generalisation and adaptability to varying linguistic patterns. However, it can not handle very long contexts and requires substantial labelled data. Vanishing gradients are another potential disadvantage which could occur over a very long sequence. Despite these disadvantages, the LSTM approach remains a strong baseline and often outperforms simpler models(CRF) in capturing semantic nuances for the anonymisation task.

BERT: A pre-trained transformer model that can be fine-tuned for various NLP tasks, including text anonymisation. It uses bidirectional self-attention mechanisms, which help to find the information which should be anonymised (Vaswani et al., 2017). This architecture ensures semantic understanding of the input and allows for classification of PII. However, Bert is computationally heavy due to the number of parameters; also, the architecture of the model and its masked language modelling architecture could lead to privacy concerns, since the model may memorise sensitive data.

ELECTRA: Compared to BERT's "Masked Language Modeling" (Salazar et al., 2019) this model uses approach called "Replaced Token Detection"(Clark et al., 2020). The approach trains to distinguish real tokens(PII) from replacements(NoPII)(Catelli et al., 2021). ELECTRA's token-level discrimination architecture improves its ability to detect sensitive phrases. However, it is less studied in the anonymisation context and its reliability heavily depends on fine-tuning strategies(Catelli et al., 2021).

In addition to Rule-Based and traditional Machine Learning approaches, there are some solution for anonymisation tasks which utilises LLM models. In Below paragraphs, we will provide a brief description of LLMs' approaches for the text anonymisation task.

Custom Transformer. A custom transformer specifically designed and

trained for the text anonymisation task (Asimopoulos et al., 2024) provides greater flexibility. Using the model architecture and training objectives directly for anonymisation can enhance performance on sensitive entity recognition compared to general-purpose models. A disadvantage of this model is the need for extensive labelled data and careful architecture selection to avoid overfitting and to maintain generalisability.

Microsoft Presidio Model. Developed by Microsoft for text anonymisation tasks (Kotevski et al., 2022), this model provides robust performance in detecting and masking various types of sensitive information. It combines predefined rules with machine learning-based components, which can sometimes limit its adaptability to ambiguous data. Nevertheless, due to its reliability, modularity, and ease of integration, we use Presidio as our PII tagger at the preprocessing stage and as a baseline in our approach. More information about the setup of Presidio for our approach will be provided in Chapter 4.

The authors of one of the recent text anonymisation benchmarks (Asimopoulos et al., 2024) claimed that the models above as state-of-the-art approaches. They fine-tuned all of them on (Sang & De Meulder, 2003) dataset since it is robust and diverse. To evaluate the performance of the models, the following metrics were used for comparison: precision, recall and F1-score.

After comparing the results of the models of the modern and traditional approaches, the authors came to the following conclusions. Fine-tuned models have shown promising results with an F1 score from 0.7 to 0.95. Among modern approaches custom transformer model demonstrated the best result. Authors also mentioned that traditional techniques like CRF and LSTM still demonstrated competitive performance with F1 scores equal from 0.76 to 0.93. All of the approaches were evaluated on the TAB dataset.(Pilán et al., 2022)

Both of the approaches have their own advantages and disadvantages.

Advantages of Hand-Crafted Approaches:

- Ability to handle direct identifiers like names, locations reasonably well using named entity recognition (NER) models.
- Simplicity and interoperability due to rule-based logic, easier debugging and compliance verification.

Disadvantages of Hand-Crafted Approaches:

- Struggle with context and semantic inferences needed to identify quasi-identifiers
- Limited ability to balance privacy protection and data utility preservation, resulting in overmasking or under-protection.
- Heavy reliance on rules/dictionaries, which may not generalise well across different domains.

Advantages of LLMs approaches:

- Contextual understanding, allowing more accurate identification of implicit and quasi-identifiers.
- Ability to generalise better across different domains and linguistic variations(higher precision and recall on diverse datasets).

Disadvantages of LLMs approaches:

- Lack of transparency and interpretability.
- Potential for memorisation of training data, which could lead to data leaks if not properly managed.
- Dependency on large labelled datasets, which can be costly to create and also cause bias.

Despite the large and diverse number of approaches to anonymising text data, there are still difficulties in this area. One of the key questions in the field is how to qualitatively evaluate the success of anonymisation models and which metric to choose for this task.

Among other difficulties, the (Patsakis & Lykousas, 2023) note challenges like: lack of balance between disclosure risk and data utility, handling diverse types of personal information, robustness against evolving privacy attacks.

Also, in an anonymisation task, the kind of data to be anonymised is critical. Neural approaches could be more suitable in the case of capturing contextual nuances as it stated in (Asimopoulos et al., 2024). despite the rapid development of neural methods in the tasks of anonymisation, traditional methods still maintain high performance. However, in any case, the choice of approach will always depend on the specific task and most often it is a search for a trade-off between anonymisation and preservation of useful data.

2.4 Existing approaches for speech

Speech data introduces a new level of complexity, so most speech anonymisation work is aimed at how to change the voice, and not at how to hide personally identifying words. Emotions of the speaker could also be considered as an identifier in some works (Tomashenko et al., 2024).

In this section, we provide an overview of existing audio-based approaches in the field of voice anonymisation. Most voice/speech/audio anonymisation methods primarily focus on masking or obfuscating voice characteristics—such as pitch, timbre, and rhythm - to conceal speaker identity. However, our research differs from this focus, as we do not focus on voice characteristic anonymisation but on classifying the PII content inside based only on audio features input. This section aims to highlight that, despite superficial similarities between text and voice anonymisation tasks, the core conceptual definitions differ significantly. Voice anonymisation must address complex acoustic properties and speaker-dependent features in audio signals, whereas text anonymisation remains around explicit textual content(But we also mentioned before that, for example, a particular writing style could be an identifier). This section will highlight a lack of solutions which are focused on the content of the audio instead of its sonic characteristics.

One work in which only characteristics of voice are considered as an object that should be hidden was written by (Jin et al., 2009). The techniques which the authors used during their work are called Voice Conversion Technique(Iglesias, Del Carmen et al., 2019). This technique is based on the spectral parameter trajectory to transform the voice characteristics of speakers. Authors also proposed a technique called "transerpolation" which combines transformation and interpolation of voice characteristics(firstly, the original speaker's voice aligned with a standardised voice model (transformation), followed by blending it with the distinctive vocal characteristics of a chosen target speaker).

Among other things, the task of voice anonymisation also involves neural network methods. Some of the most recent of them are described in (Panariello et al., 2024) paper. The principal difference between the proposed anonymisation method and others lies in the use of the neural audio codec(NAC). Authors utilise NAC as an encoder and decoder for audio data.

NAC - a sound compression method that uses neural methods to transform audio signals. (Défossez et al., 2022). They use an encoder–quantizer–decoder architecture: the encoder compresses raw au-

dio into lower-dimensional embeddings, the quantizer discretises these embeddings into codes, and the decoder reconstructs high-fidelity audio from the codes (Défossez et al., 2022). This architecture makes NACs particularly effective for voice anonymisation tasks, as they reduce identifiable speaker characteristics while preserving intelligibility and naturalness.

Table 2.1: Technical Comparison of Voice Anonymisation Methods

Method	Description
Sequence-to-Sequence Conversion	Voice Utilises sequence-to-sequence (seq2seq) models that leverage pre-trained speech and speaker embeddings to learn a direct mapping from original speech representations to anonymised ones. This approach focuses on disentangling speaker identity from linguistic content by transforming speaker-related features while preserving phonetic information. Challenges include maintaining speech naturalness and intelligibility post-conversion.
Neural Audio Codec Language Models	Combines neural audio codecs (NACs) that quantise speech into discrete codes with language models trained to generate anonymised speech. NACs act as a bottleneck to compress and obscure speaker identity information effectively. Evaluations based on the VoicePrivacy Challenge framework demonstrate this method's efficacy in balancing voice anonymisation and speech quality, although synthesis artefacts and latency remain concerns (Tomashenko et al., 2022).

We can see that there are solutions for anonymising speech, namely, masking the voice characteristics of the speaker, but this does not take into account the content of the speech itself, namely, what words the speaker utters. This could lead to the disclosure of a person's identity using various identifiers used by the person.

This problem was mentioned and widely discussed in the most recent paper regarding voice anonymisation by (Champion, 2024). In this paper, the author describes not only existing solutions for voice characteristic anonymisation, which we also touched on before, but also mentions

the lack of a universal solution which can anonymise both audio characteristics and the content inside of it. Moreover, the author also pointed out the weak points of existing solutions and developed a new attack method to invert anonymisation and tested it on a quantisation-based transformation method. The method itself was proposed by the author as an innovative one.

The closest anonymisation system to the one that we will try to implement in our research is ASR + NER de_ID approach proposed by (Cohn et al., 2019a). Authors of this approach transcribe audio to text, finding PII with NER methods, and then aligning spotted PII with audio input and cutting and replacing this PII with silence. However, the main shortcoming of such an approach is transcription, which is not always available for audio data and itself contains a lot of Personal Information. A more detailed description of their approach will be provided in Chapter 4 as we will use their logic as an inspiration for our approach.

A similar approach to the one described in (Cohn et al., 2019a) has been proposed by the commercial company Aiola.AI³. Their method essentially replicates the core idea from (Cohn et al., 2019a), with the main difference being the ASR model they employ. Aiola.AI first transcribes audio into text with Whisper and then applies an NER model to extract the desired information ⁴. The model they use is open-source, and according to the project’s webpage, it is intended to help protect “private and sensitive data.”⁵

Their system supports open-type NER, allowing it to recognise a broad range of entity types, including previously unseen categories at inference time. The authors claim that it achieves performance comparable to or better than strong pipeline baselines in both transcription accuracy (WER) and entity-recognition metrics (F1 score).

However, several limitations are also mentioned. The model is trained on large-scale synthetic speech paired with annotations generated by a large language model, which may introduce systematic biases and compromise evaluation independence. Fine-tuning on synthetic speech also leads to a slight reduction in transcription accuracy relative to the original Whisper model. Furthermore, the joint ASR-NER approach requires careful handling of training strategies—such as entity-type dropout and negative sampling—which increases training complexity.

Although this approach is feasible and, as Aiola.AI suggests, could be

³<https://aiola.ai/>

⁴<https://github.com/aiola-lab/whisper-ner?tab=readme-ov-file>

⁵<https://aiola.ai/blog/whisper-ner-model-for-enhanced-data-privacy-security/>

adapted for audio anonymisation tasks, it still fundamentally relies on transcription.

Thus, we see a lack of a solution for de-identifying speech content from audio data, and this will be the main direction in our research work.

2.5 Gaps in the Literature

The Literature Review section provides descriptions of current methods of anonymisation systems, both for text and audio. However, we can see a lack of approach which focuses not on voice characteristics but on the context of the speech and does not rely on any transcription.

Therefore, we can see a clear gap in the research field of the PII classification task on only audio features. Our proposed approach in Chapter 4 for the PII classification task with only audio features could become a base for future works in this direction, and different pooling techniques could help to determine the strong and weak sides of it. The anonymisation approach proposed by the private company above also indicates that there is a growing demand for audio anonymisation methods. It highlights the need to explore solutions that do not depend on transcription and instead focus on anonymising the spoken content itself, rather than the speaker's characteristics.

Chapter 3

Motivation and Approach

3.1 Motivation and Challenges

Having examined and compared text and audio approaches, we can come to the conclusion that in the audio segment, there is less work and it is focused more on anonymising the characteristics of the speaker's voice than on the content of the audio itself. Therefore, in this work, we will try to develop audio only solution for the PII classification task.

In this research work, we will focus on the PII classification task first. The goal of our potential approach is to check whether it is possible to classify PIIs with Encoder-based models and only audio input.

The main challenge for our approach is the lack of properly annotated data. Therefore, a lot of discussion in this work will be dedicated to the annotation and labelling of PII parts for the proposed approach. More information about our dataset, its structure, and the methodology of its annotation and labelling will be provided in Chapter 4.

The creation of such an audio solution for PII classification in audio recordings in future can allow the use of audio voice recordings for research purposes or for the development of products that could guarantee the privacy and anonymity of their users in the audio domain.

3.2 Task formulation and possible solution

In this work, the task is to attempt to explore possible solutions for the anonymisation of audio recordings. Examine how good modern encoder-based models can perform within the task of PII classification, what is the bottlenecks and how it possibly can be resolved.

A possible solution for working with audio is a compilation of existing approaches for text plus speech recognition and synthesis models for processing input and generating output. As an input, such a system will receive an audio recording with a voice, and then decipher it using the speech-to-text model (this could be the Whisper model as state of the art in its domain). We apply one of the anonymisation models to the received text, which generalizes the identifiers present in the text. At the last stage, we synthesize the speech back using the text to speech model. Something similar was proposed in (Cohn et al., 2019a) paper where author used similar schema for NER task and then regenerates audio.

This solution could be a possible one, but there are some possible disadvantages which worth considering in advance of this approach. First of all, the audio regenerated output may lose the intonation of the original recording as well as the original tempo. In addition, it is important how accurately the data will be transferred from audio to text format. Although the large Whisper model demonstrates excellent results in terms of the Word Error Rate (WER) metric (especially for English language), it is still not 100% accurate. Moreover, the data that will be received after the speech-to-text process may differ greatly from the data on which text anonymisation models are trained, and in this case, we will have to mark up the data manually and fine-tune own model on this data specifically for a text transcribed by Whisper(or any other STT model).

In addition to creating a ready-made solution, an interesting direction of work may also be an attempt to study the voice characteristics themselves in the task of speech anonymisation. Perhaps while a person utilises PII, which is related to them, voice characteristics change in such a way that this can be caught and used as features for training. For example, we can mark out all the time intervals when a person says his name and look at the tone of intonation, etc., which is different from the rest of the speech.

There are a lot of interesting directions for research in the field of speech anonymisation with only audio features, but we will mainly focus our research on the attempt to create a system which operates only on audio input features to classify certain segments of audio as PII and NOPII. This narrow scope will allow us to focus on the question of whether the task of

speech anonymisation is possible at all with audio encoder-based models.

3.3 Datasets

The dataset for this work can be any audio data source in which the speaker names personal data. The interview format best fits this description.

Anyway, almost any audio corpus of dialogues can be used for the given task of anonymisation; it is only important that it contains sensitive information that can be considered as an identifier. The corpus itself can be further pre-transcribed using Whisper, and then possible errors can be manually edited during the transcription process.

The main difficulty associated with the data is the lack of a gold standard for training the model, so most likely it will be necessary to manually annotate some amount of audio recordings in order to get a gold transcription for the baseline.

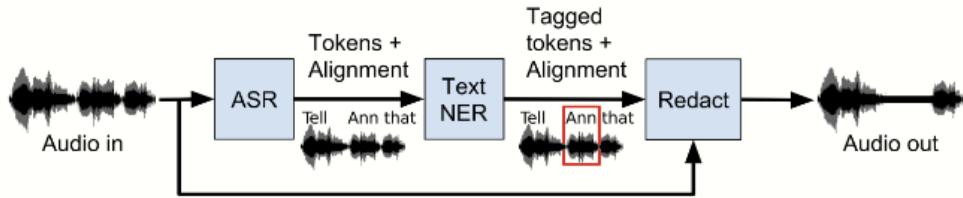
In general, for the task of PII classification with only audio features, we require to have: audio interviews(which will further be used as a source of features for training our classification model) and transcription of those interviews (this transcription will be used for PII tagging. Each word in the transcription has its timestamps, and the anonymisation text classification system will be used for these words to further label them. This way, we will know when and what happened in the audio data and then pass this information as labels together with audio features to our models for the training process).

More details about a particular dataset, its preprocessing, annotating and labelling will be provided in Chapter 4.

3.4 Approach

For researching the possibility of models to classify audio sequences as an identifier or not, general approach which will be used throughout the research work was developed.

Figure 3.1: de-ID Pipeline from (Cohn et al., 2019a)



In Figure 3.1 above, we can see the proposed approach schema for speech anonymisation in the paper of (Cohn et al., 2019a). In their work, authors take an audio input and then transcribe it into text; after that, they use an NER model for PII tagging and align the words and tags with audio time stamps. After that, they cut out the PII segment from the audio. Our approach is mainly inspired by this work, but with one significant difference, that our anonymisation part will not rely on text transcription, but will be based on audio features of the audio input itself.

Figure 3.2: Figure of Approach

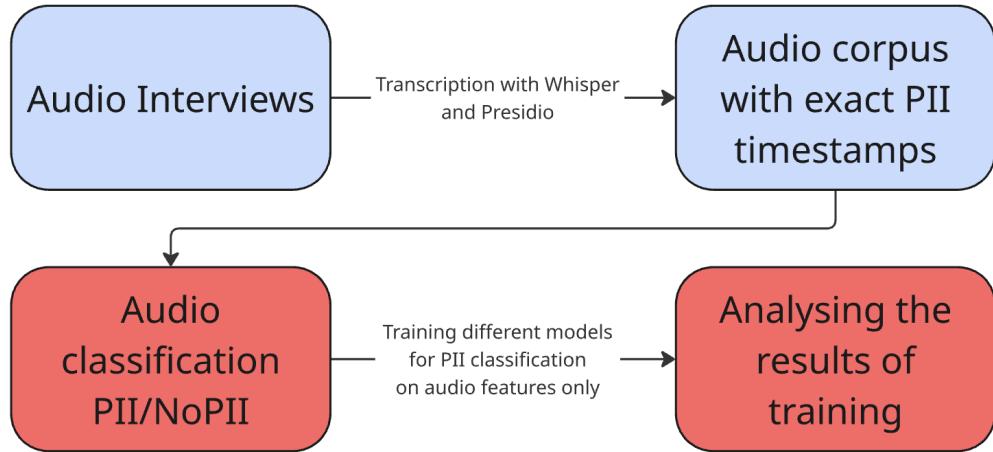


Figure 3.2 shows the general pipeline for this research work. The main difference with the approach proposed in (Cohn et al., 2019a) is that the transcription part is only required in the preparation part(blue boxes) of this research and is not required for classification once the model is trained(red boxes).

The goal of our approach is to create an audio classification model for PII detection, which receives as input an audio piece and as an output

provides a label for this audio piece. This audio-only classification system can further be extended to an anonymisation system(for example, we will cut audio pieces which were classified as PII from the input audio, likewise it was done in (Cohn et al., 2019a) paper).

In this research, as a main tool for PII detection, we will use 3 pretrained models(WavLM, Hubert, and Whisper; More information regarding the model architecture and why those models were selected for this particular task will be provided in the relevant Section 4.7).

For each model will be conducted several sets of experiments. First of all, two approaches **binary** and **multilabel**. In binary approach models classifies audio sequence within two general categories: PII and NoPII. Meanwhile, in multiple-label models, they are challenged to classify audio sequences with exact labels (PERSON, NRP, LOCATION etc.). More information about labels for multilabel approach could be found in section 4.3.

For each of the models and approaches will be performed with different size of input. More detailed description of each input window and how context is added to input will be provided in Subsection 4.4.2.

And last thing, which will be point of experiment will be a different representation aggregation technique or pooling techniques. In this research paper, we will focus on 3 particular techniques: gated pooling, attention pooling, and hierarchical pooling. As well as more information about each technique and why they are important in this research will be provided in Sections 4.5.

To sum up, in this research work, in order to determine machine learning abilities to classify only audio data input as an identifier or not, 3 models, 2 approaches, 4 input sizes, and 3 pooling techniques and context/no-context features were used. In general, 144 experiments should be conducted, which will help to determine the behaviour of models and figure out what is important in the task of only audio PII classification/anonymisation task.

3.5 Evaluation

A critical component of this research is the evaluation of our PII classification approach. At its core, the task we address is a classification problem, where we explore two principal strategies for PII categorisation: binary classification (distinguishing PII from non-PII) and multilabel

classification (assigning PII types to their respective classes).

Our evaluation metrics should aim to quantify the performance of both strategies, employing established classification metrics to assess how well they identify and categorise sensitive information.

By examining metrics such as precision, recall, F1-score(F1 micro and micro), AUC(micro and micro), mAP we can comprehensively measure the trade-offs between false positives and false negatives, ensuring that our models effectively balance privacy protection with data utility. This simple evaluation framework allows us to understand the strengths and limitations of each classification category within the overall PII detection system and will help to answer main research question: if PII classification with only audio features possible at all.

More detailed description of each metric and how it will work in relation with our data will be provided in Section 4.6 of Chapter 4.

Chapter 4

Setup and Methodology

After we have established the approach in Chapter 3, we can proceed to a detailed examination of the methods and tools used for the implementation. In the following sections, we provide comprehensive descriptions of the encoder-based models, dataset characteristics, input size considerations, context handling strategies, pooling techniques, hyperparameter configurations, and evaluation metrics employed in our study.

4.1 Data selection

Selecting a dataset for an anonymisation task is a key step in work of this kind.

A possible dataset for experiments could be a corpus of interviews collected as part of The New York Public Library's Community Oral History Project¹. The dataset consists of 8 audio recordings with a total length of approximately 8 hours. Each audio recording is an interview with a New York resident who tells a personal story. The importance of this dataset in particular for the anonymisation task is that such records contain a lot of personal information about the speaker. The disadvantage of this dataset is that it does not have available text transcription, but it will be needed in any case (either to evaluate the effectiveness of the approach or for speech synthesis)

Another possible dataset for this work could be the CallHome corpus (Canavan et al., 1997). The corpus consists of 120 unscripted telephone conversations between friends/family members. All conversations are

¹<https://www.kaggle.com/datasets/audreyfeldroy/oral-history-audio-interviews>

no more than 30 minutes long and each conversation has a very detailed markup of a 10-minute excerpt from the entire audio file. This corpus is of particular value because it has a very detailed transcription of dialogues, containing not only the linguistic content of the dialogue, but also the speaker's turn, and specific time systems for each phrase spoken by the speakers.

It is important to mention, a dataset for this type of work must contain a lot of identifiers(personal details like name, address, religion, political views, etc), and this criterion creates difficulties in selecting data. The main difficulty in selecting data is that such corpora are usually non-public precisely because of the data they contain. For example, mentioned earlier (Canavan et al., 1997), despite having real conversations not have that many identifiers because often people in these interviews and dialogues discussed not their lives, but some other events. It is not stated anywhere on any of CALLHOME resources, but in one of the interviews, the participant mentioned that according to the guidelines, they are forbidden to talk about topics related to them, precisely to avoid data disclosure. Thus, this corpus is not suitable since these dialogues were ideologically designed not to disclose the personal data of the participants.

The spoken Wikipedia corpus collection (Baumann et al., 2018) could also be a potential dataset for the anonymisation task, but this corpus has its drawbacks. Despite the fact that the Wikipedia corpus itself implies the presence of a large number of identifiers and personal information (for example, biographies of scientists, politicians, etc.), it is not a product of interaction between two people. If a model were trained on such data, there is a risk that the trained model would not be suitable for data from other domains and, in particular, for dialogues. In addition, these Wikipedia articles are recorded by speakers, which makes the speech unnatural compared to normal dialogue, and the idea of using audio features solely for recognising and removing identifiers loses its meaning since real dialogue differs from the speaker's speech.

After studying these two corpora, it was decided to look towards sociolinguistic interviews, more specifically *narrative* ones, which have been constructed for qualitative research methods. Narrative structured interviews are a method that combines elements of structured interviews with narrative approaches (Moenandar et al., 2024)(Talmy, 2010). These interviews are interesting to this particular research from the point of view that most likely in them the respondent, answering questions, will mention many details about their life, such as place of residence, place of work, native language, etc. The structure of the narrative interview involves questions of the following type:

- Who are you?
- Where are you from?
- What is your native language?

Basically, structured narrative interviews are just interviews about a person's background and life experiences related to the research topic. and it seems like The New York Public Library's Community Oral History Project² is a perfect match for the anonymisation task. Since the purpose of these interviews is to talk to New Yorkers and hear their stories, it is expected that a lot of answers involve information regarding the participant's background, place of work, family, university, etc.

Although this data seems ideal for working with, there is a problem in that only 9 interviews are available for downloading from Kaggle ³, and getting the rest is a separate challenge.

4.1.1 Data crawling

To work with data from The New York Public Library's Community Oral History Project, it was necessary to download the interview from the project's website.

In order to download the data we wrote a script using Python and its libraries.(Selenium module + Google web-driver) The script iterated over the project's page and downloaded audio recordings from each of the web-pages. To avoid blocking, a random delay of 3-5 seconds was set between download iterations. The delay allowed downloading all the audio from the website without failure in a little more than 2 hours.

²<https://www.nypl.org/digital-research/projects/community-oral-history-project>

³<https://www.kaggle.com/datasets/audreyfeldroy/oral-history-audio-interviews>

Figure 4.1: Distribution Figure

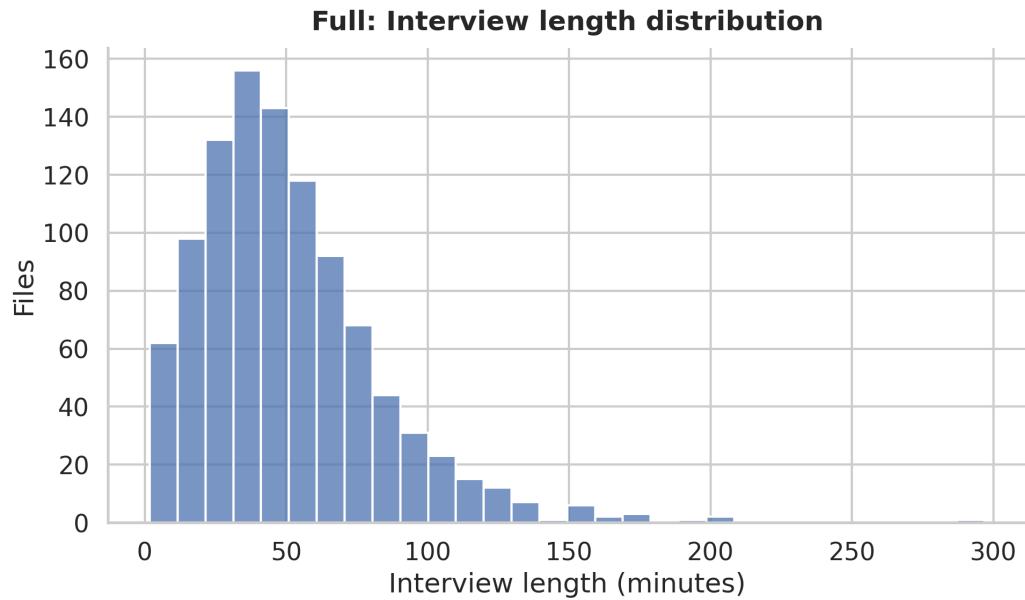


Figure 4.1 shows the length of downloaded files. Overall, the dataset contains one thousand seventeen audio files, ranging in length from fifteen minutes to two hours on average. The next step before training the model was preprocessing, which will allow us to use this data for model training.(audio transcription(STT), statistics collection, analysis, silver corpus preparation)

4.1.2 Data pre-processing

The audio files with the interviews themselves have no particular value for our task. In order to train the model to predict where in audio an identifier occurs, the input needs to include timestamps of PII in addition to raw audio features. However, after crawling the data, we have only audio files without any transcription. Therefore, the new challenge occurs.

To achieve the goal of a proper dataset for this task, it was necessary to extract information about identifiers and when they occur. Of course, this task could possibly be done with manual annotation but it is really time-consuming. There are various tools which we can combine to achieve meaningful and reliable annotation for the speech anonymisation task.

In order to achieve that dataset, we will need two main tools, Whisper and Presidio, each of which are perfect in its domain. (fix formulation)

Whisper is a state-of-the-art speech recognition model (Radford et al., 2022) developed by OpenAI. It produces not only the transcription of the speech but also provides us with punctuation and very precise timings, which will be very important in the future for training a PII classification model. A more detailed description of this model could be found in Section 4.7, which describes the model's architecture in detail.

Presidio by Microsoft is a tool for spotting identifiers in text data. This particular anonymisation API was chosen because it provides us with a simple set of labels, it's relatively easy to use and is considered a reliable and widely used approach in the task of text anonymisation. However, it is worth mentioning that Presidio is not considered a state-of-the-art solution according to the most recent survey on text anonymisation task (Deußer et al., 2025b). For text data, it remains a highly trusted tool which could be used for our dataset preparation task.

Figure 4.2: Pipeline for dataset preparation with Whisper and Presidio

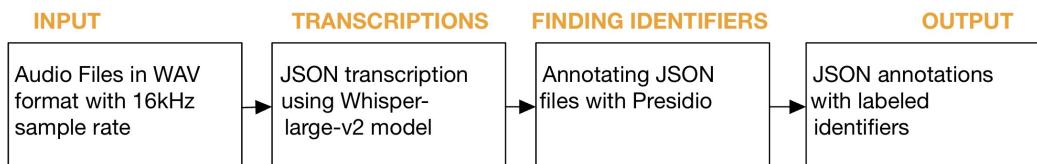


Figure 4.2 demonstrates how audio data was processed into a JSON transcription with exact time stamps and labels for each word.

At first, we performed basic preparation of audio data and made sure that all of the data are in proper Whisper-ready format, which is 16kHZ WAV audio files. Another thing that was considered is to lowercase all of the file names and make sure that the folder contains only audio content.

After the audios were prepared, the process of transcription was initiated. After processing audio with Whisper-large-v2, JSON transcriptions with exact timestamps for each word and phrase were obtained.

Those transcriptions were further passed to the Presidio Analyser, which automatically annotated each phrase and word with labels. Presidio was initialized using its default configuration, which supports a wide range of entity types such as names, locations, dates, organizations, and numerical identifiers (phone numbers or credit card numbers etc. List of all will be provided in Section 4.3). Each word in the Whisper transcription output was analyzed individually using the AnalyzerEngine, which applies a combination of pattern-matching rules, regular expressions, and pre-

trained Named Entity Recognition (NER) models to identify potential PII. Detected entities were labeled directly in the JSON transcription file, and their corresponding timestamps were extracted for subsequent processing.

As a result, more than 1000 audio files obtained automatic JSON transcription with labelled identifiers and exact time stamps for each word, which could be further used for training models for the anonymisation task.

It is also worth mentioning that automatic speech recognition and automatic audio transcription is very computationally demanding tasks and require not only computing power, but some time to process. Average time for 1 audio file to be processed was equal to 15 minutes, which means that total transcription time was about 300 hours. This constitutes an important result of our work

However, this dataset is obviously not ideal, and it is worth considering that some of the results should be viewed with caution. As the transcription may have imperfections due to being a Silver Standard corpus.

Concept of Silver Standard corpus is widely used in many of research works. Silver Standard refers to a dataset where annotations are not manually created by a group of experts but created by combining outputs from multiple annotation systems. Example of how Silver Standards created could be seen in paper (Schuhmann et al., 2010), where Authors used 4 different NER taggers for creation of their CALBC corpus.

4.2 Silver Standard, Dataset Structure and Insights

After all the manipulation with data, as a result, Silver Standard/dataset was obtained.

Since the training process model will work only with both audio and text input, it is important to properly organise the data. It was decided to have two datasets: general datasets for experiments, one small demo with 33 audios and 33 transcripts(to test approaches and make sure that setups are sane) and one big dataset for full training with 1017 audios and 1017 JSON transcriptions, which will be used at the end after we get results for models with the demo dataset.

Table 4.1 has summaries about two datasets that are used in this work. We can see that both sets have relatively the same balance in terms of

Table 4.1: Dataset summary: Full vs Demo

Metric	Full	Demo
Total files	1017	33
Overall seconds	3,100,125.22	88,502.46
Overall hours	861.15	24.58
Avg. interview length (min)	50.81	44.70
Avg. identifiers / file (word)	278.21	309.15
Avg. identifiers / file (phrase)	157.27	136.33
Total words	7,505,956	213,498
Total identifiers	282,942	10,202
Avg. word length (chars)	4.23	4.20
Avg. word duration (sec)	0.3260	0.3256
Identifiers per word (%)	3.77	4.78

word length, which is equal to approximately 0.32 seconds with about 4.2 characters per word. The information about word length will be further used for deciding on the input size window for one of the classification approaches. We can also see that the concentration of labels is higher by 10% in the demo set and which could be better at the demo part of the project since the demo set is more balanced. Interview length is almost the same and equal to 44 and 50 minutes for the demo and full dataset, respectively.

Statistics from the table confirm that the demo corpus has comparable characteristics in terms of lexical and temporal properties to the full one. While being significantly smaller than the bigger corpus, it is PII-wise richer in terms of distribution and amount of identifiers, which makes it solid for prototyping and testing before full-scale training.

For a deeper understanding of the dataset structure and in order to understand how labels are assigned to the words with Presidio, two examples of annotation will be examined in the following paragraphs.

On the Listing 4.1, you can find an example of the inner content of each JSON transcription.

Listing 4.1: Example dataset entry

```

1  {
2      "phrase": " Aiden, you're 35 years old and you've lived in
3          Harlem all your life. Tell me some of your",
4      "start": 22.84,
5      "end": 28.52,
6      "identifier": 1,
7      "identifier_ph": [ "PERSON" , "DATE_TIME" , "LOCATION" ],

```

```

7   "words": [
8     {"word": "Aiden,", "start": 22.84, "end": 23.32, "identifier": 1, "labels": [ "PERSON" ]},
9     {"word": "you're", "start": 23.42, "end": 23.5, "identifier": 0, "labels": []},
10    {"word": "35", "start": 23.5, "end": 23.84, "identifier": 1, "labels": [ "DATE_TIME" ]},
11    {"word": "years", "start": 23.84, "end": 24.14, "identifier": 1, "labels": [ "DATE_TIME" ]},
12    {"word": "old", "start": 24.14, "end": 24.46, "identifier": 1,
13      ...
14      {"word": "in", "start": 25.64, "end": 25.76, "identifier": 0, "labels": []},
15      {"word": "Harlem", "start": 25.76, "end": 26.08, "identifier": 1, "labels": [ "LOCATION" ]},
16      ...
17      {"word": "your", "start": 28.4, "end": 28.52, "identifier": 0, "labels": []}
18    ]
19 }

```

The transcription itself is a sequence of transcribed phrases with exact time stamps when this phrase occurs. In addition to time stamps, Presidio marks the whole phrase with a tag identifier and makes it equal **1** if any identifiers occur and **0** if not. Also, if phrase contains several identifiers, Presidio will add them to the list of identifiers in tag **identifier_ph**.

In addition to phrase-level transcription, each phrase is divided into words. As well as phrases, each word has information about exact time stamps, tag identifier with binary value, and tag labels which have exact label value for every word if a word is identifier (for instance PERSON, LOCATION etc.)

In this particular Listing 4.1, Presidio managed to identify all PII correctly. However, the perfect annotation is not always the case, especially when it's automatic. In a Listing 4.2, we can clearly see that Presidio confuses PERSON and LOCATION tags in line 11.

Listing 4.2: Example dataset entry with multiple labels

```

1 {
2   "phrase": " and then you went up to Westchester and
3   Hastings on the Hudson.",
4   "start": 167.16,
5   "end": 170.76,
6   "identifier": 1,
7   "identifier_ph": [ "LOCATION" , "PERSON" , "LOCATION" ],
8   "words": [
9     ...
10     {"word": "Westchester", "start": 167.88, "end": 168.62, "identifier": 1, "labels": [ "LOCATION" , "PERSON" ]},
11     {"word": "and", "start": 168.62, "end": 169.16, "identifier": 0, "labels": []},
12     {"word": "then", "start": 169.16, "end": 170.76, "identifier": 0, "labels": []}
13   ]
14 }

```

```

10   identifier": 1, "labels": [ "LOCATION" ]},
11   {"word": "and", "start": 168.62, "end": 169.76, "id
12   identifier": 0, "labels": []},
13   {"word": "Hastings", "start": 169.76, "end": 170.2, "id
14   identifier": 1, "labels": [ "PERSON" ]},
15   {"word": "on", "start": 170.2, "end": 170.36, "identifier
16   identifier": 0, "labels": []},
17   {"word": "the", "start": 170.36, "end": 170.48, "id
18   identifier": 0, "labels": []},
19   {"word": "Hudson.", "start": 170.48, "end": 170.76, "id
20   identifier": 1, "labels": [ "LOCATION" ]}
21 ]
22 }

```

In the examples above, we can clearly see that the label annotation is not perfect. However, a comprehensive evaluation of annotation accuracy would require manual review of each transcription, which is beyond the scope of this research. Since the main goal of this paper is to check whether it's possible to train the model to detect identifiers only on audio features, we can use this imperfect data for these purposes. The results derived from this silver standard corpus should be interpreted with acknowledging of potential imperfections and inconsistencies in automatically generated annotations.

In conclusion, for this work, two corpora were prepared: **demo** and **full**. The Demo corpus consists only of 33 audios and the same amount of transcripts for testing the approaches and finding the optimal training script. Demo corpus is divided into three smaller ones, namely: **training**, **validation** and **test**. Training, validation and test consist of 26, 4 and 3 audio-transcription pairs, respectively. The Full corpus tends to be used after finding out the best approach and training script.

Even though we assume that an imperfect silver corpus will be used for training, it remains important to have at least some perfectly annotated data; therefore, a Gold Standard is still necessary.

4.3 Gold Standard

In order to further evaluate the quality of anonymisation during this research, we will need a gold standard. To create a gold standard, it is worth determining in advance what PII we will designate and how.

Gold Standard data is required for our particular approach in order to check to which extent our Silver Standard is sane. It will also allow us

to compare our approach performance to Presidio Performance on same data. In general manual annotation serve as a tool powerful tool of approach verification.

During pre-processing, the Presidio model was used to locate the PII in transcribed audios. The authors of the Presidio model predefined 38 types of PII; among them, 12 are general types, and the remaining 26 are specific country-related types(USA, UK, Spain, Italy, Poland, Singapore, Australia, India, Finland). Since all of the interviews were recorded with New York inhabitants, we can exclude all other countries except the USA from the long list of useful PII. Table 4.1 below shows both general and USA PII in, which represent a long list of identifiers for the annotation guidelines.

Table 4.2: General and USA-Specific PII Types

General PII Types	Description
CREDIT_CARD	A credit card number(12 to 19 digits).
CRYPTO	A cryptocurrency wallet number (Bitcoin address).
DATE_TIME	Absolute or relative dates.
EMAIL_ADDRESS	Email address of the person.
IBAN_CODE	The International Bank Account Number (IBAN).
IP_ADDRESS	An Internet Protocol (IP) address that identifies a device on a network.(IPv4 or IPv6)
NRP	Information regarding nationality, religion, or political views of the person.
LOCATION	Any names of cities, provinces, countries, rivers, mountains etc.
PERSON	First name, middle name, last name, of the individual.(including initials)
PHONE_NUMBER	A telephone number of the person.
MEDICAL_LICENSE	Medical license numbers assigned to healthcare professionals.
URL	A web address that points to a specific resource on the internet.

USA PII Types	Description
US_BANK_NUMBER	A US bank account number(8 to 17 digits)
US_DRIVER_LICENSE	A US driver license number.
US_ITIN	A US Individual Taxpayer Identification Number (ITIN)(9 digits)
US_PASSPORT	A US passport number(9 digits)
US_SSN	A US Social Security Number (SSN)(9 digits)

While Presidio provides a wide range of PII, not all of them apply to our audio data and are unlikely to be found in it. Identifiers that are unlikely to be encountered and are unlikely to be useful when studying audio data include CREDIT_CARD, CRYPTO, IBAN_CODE, and IP_ADDRESS, MEDICAL_LICENSE. In addition to these PII, it is also unlikely to encounter any country-related PII such as US_BANK_NUMBER, US_DRIVER_LICENSE, US_ITIN, US_PASSPORT, US_SSN. Since the

Presidio tool was designed for text, it includes such 'digits' PIIs.

Nevertheless, working with audiodata, we rarely or almost never will face such kind of identifiers, and there is no sense in keeping them in a shortlist of possible identifiers. To support this statement, the distribution of identifiers was calculated.

Figure 4.3: Distribution Figure

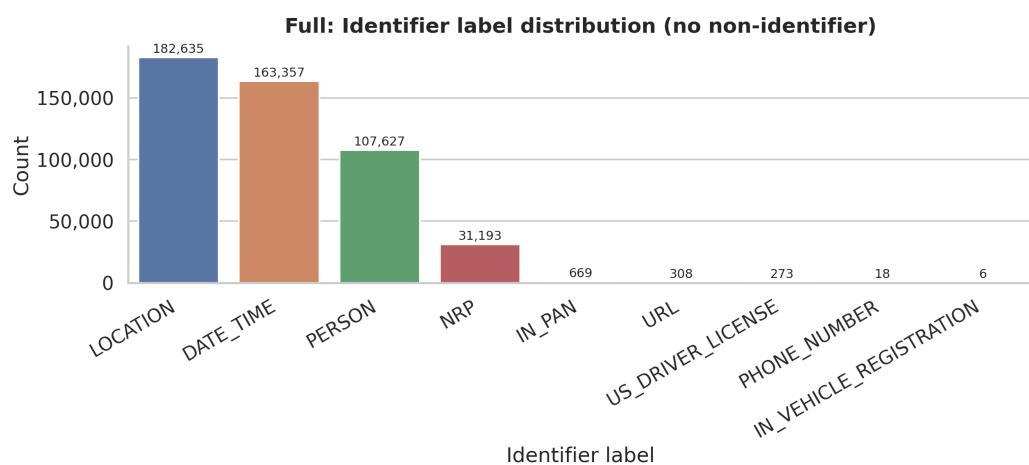


Figure 4.3 shows label distribution among all transcribed JSON files. In this figure, non-identifiers are excluded, but their amount is significantly higher and equal to 7,326,716. However, from the figure, we can see that we have 4 most frequent classes: LOCATION, DATE_TIME, PERSON, and NRP. As previously stated, digit-wise identifiers are not really presented in this particular dataset; therefore, in the interest of a more balanced representation of labels and to make the task less confusing for the model, it was decided to keep only the four most represented classes.

All classes except the four most represented were relabeled with a non-identifier tag, so 1274 words became non-identifiers. Since the training set now has only 4 identifiers, it is logical to have 4 main classes for manual transcription and for achieving golden standard transcription. The short list of PII labels for the gold standard and silver, as well, looks as follows: **DATE_TIME, NRP, LOCATION, PERSON, NONIDENTIFIER**.

After the list of identifiers was established, one audio interview was manually annotated with those labels in order to have at least one audio interview as a gold standard. This gold standard will be used in the future for baseline and further comparison with other models to the baseline. (More information about the baseline provided in Section 4.4.1)

During the annotation process, we have not used any particular guidelines for annotation. In any case, there were a couple of points that we paid close attention to. First of all, we were checking words which were already marked as PII in order to verify that the model does not confuse labels, like it was in Listing 4.2 example with Name and Location labels. Another important decision we made when annotating the data was to consider educational institutions as LOCATION - names of schools, institutes, and universities. Since our corpus is within the framework of a narrative interview, such interviews involve a lot of names of places where a person studied. However, when annotating, we noticed that Presidio often does not consider such places as PII; It would be correct to single out all such PIIs in a separate class: **EDUCATION**, but in this case, it would be an unfair comparison with the Presidio, which does not have a similar class. It is worth noting that in the future it would be better to use a wider range of classes for this corpus and to the established list of classes, at least add classes such as: place of **OCCUPATION** and **EDUCATION** (these two classes often appeared in the training sample but were rarely considered as PII).

4.4 Setup

In this section, we provide a comprehensive overview of the setup for our approach. We discuss the baseline evaluation metrics chosen for performance assessment, analyse the potential impact of different input sizes, including their advantages and challenges, explain our strategies for context handling, and describe both binary and multilabel classification configurations. Additionally, we detail the selected hyperparameter settings used to optimise the model. Models it's architectures, advantages, and disadvantages will be discussed later in Section 4.7.

4.4.1 Baseline

In Chapter 2 we already mentioned that the audio segment of the PII anonymisation task lacks a solution; thus, there is no particular existing baseline for this task. Since this research work is taking place in a new research modality, in order to achieve the first reliable baseline for the PII audio classification task gold standard was annotated.

As a baseline in this work, the Presidio model will be used. Presidio widely used anonymisation system both in research (kotevski2022evaluation), (T.-Y. Wu et al., 2022), (You et al., 2011) and previously in industry. In one

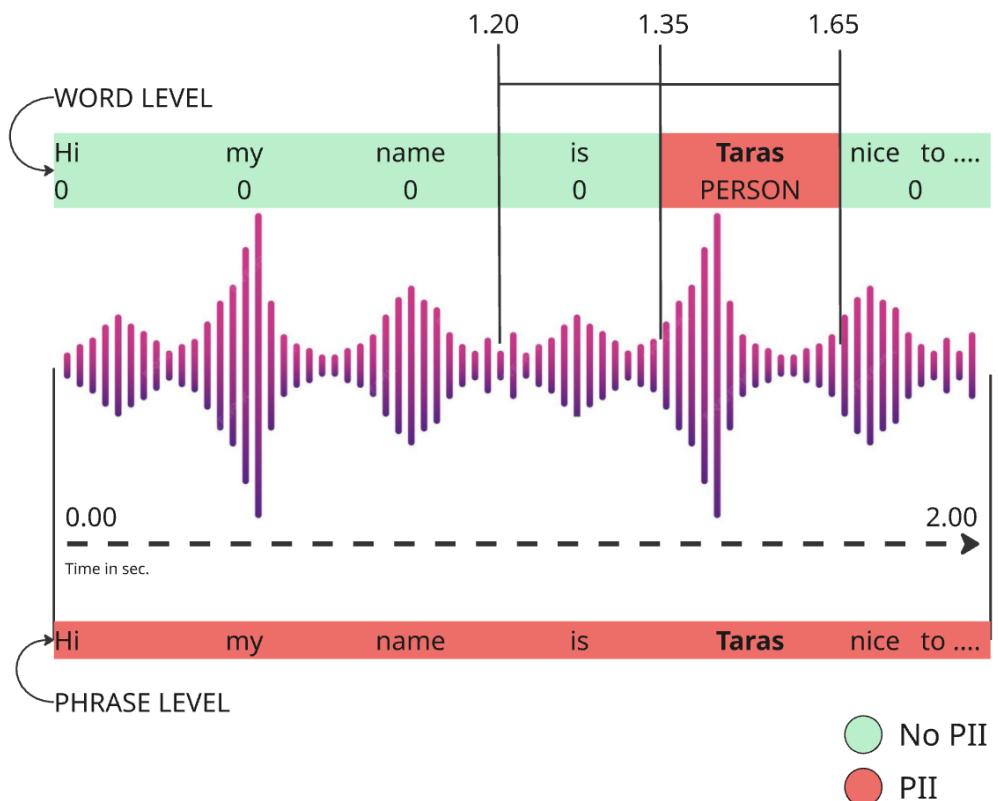
of the most recent works in the field of anonymisation of educational text data with GPT 4o (Ji et al., 2025) authors also, using Presidio as one of the baseline models.

The baseline results will be achieved by applying the Presidio model to Gold Standard transcription the resulting F1 score will represent the baseline result. (The exact results of presidio performance within the Gold standard will be provided in 5)

4.4.2 Input window

Another important part of this work is the input size or window size. In order to understand what is important during the process of PII classification from audio features, 4 different input sizes are proposed in this work. Different sizes and context setups could help to figure out context and feature importance for the classification task. Also, different sizes are important for pooling since some of the techniques require context.

Figure 4.4: Phrase and Word Level Slicing



According to the dataset structure and its transcription, we already have 2 predefined input sizes, **word-level** and **phrase-level**. Figure 4.4 has an example of those input sizes. Both these input sizes are basically dynamic sizes and represent exactly the length of the words and phrases. Neither of these input sizes requires additional alignment, since label and time stamps were predefined at the preprocessing stage, and we consider the transcription sane. However, these sizes do not take into account certain problems, such as the silence between words and phrases. Silence became not only a part of the audio features of words and phrases, but it can also be considered a valuable part and influence models' decisions. Some of the silent parts could not be considered during the training, which can potentially spoil the decision of the model. Another problem that could potentially arise, particularly with phrase-level features, is that some of the phrases contain several identifiers, and this can also confuse the model during training. Anyway, one of the advantage of word and phrase level approaches is that it will be much easier to interpret results in comparison to the other two techniques, which are equal slicing and frame level.

It seems like the prefect input size for the task of audio PII classification is word-level(it is pretty interpretable), but the main shortcoming of this approach is that the model during the inference will need to receive the exact timing for each word, which is not really aligned with the idea of only an audio anonymisation approach(since we need predefined timestamps for words for each audio, which could be obtained with Whisper, therefore it looks more like (Cohn et al., 2019a) or Aiola.ai⁴ approaches rather then independent one).

Another audio-input format used in this work is equal slicing. In this approach, each audio signal is divided into fixed-length segments of 0.5 seconds. Based on the average word duration in our dataset (Table 4.1), this window size is generally sufficient to contain at least one word. As illustrated in Figure 4.5, this method no longer depends on exact timestamps, unlike the phrase-level and word-level approaches. As a result, the pipeline becomes more independent of transcription, which is an important step toward achieving a fully audio-only speech anonymisation system.

However, equal slicing has a notable shortcoming compared to the word-level approach. With word-level slicing, timestamps from the transcription ensure that each segment contains a complete word, making the label assignment reliable. In contrast, equal slicing can - and often does - split a word across two consecutive slices, as visible in Figure 4.5. To handle this issue, we align the word-level identifier timestamps with the 0.5-second slices and determine whether an identifier is present within the

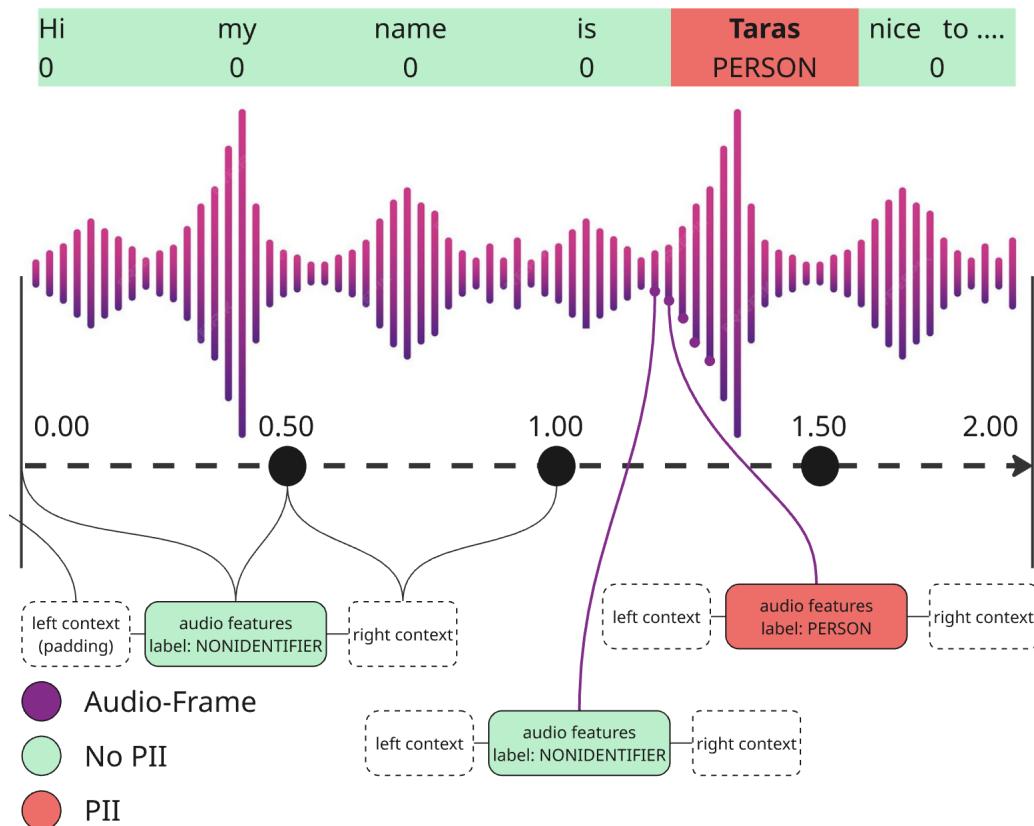
⁴<https://aiola.ai/blog/whisper-ner-model-for-enhanced-data-privacy-security/>

slice or not.

A further challenge is deciding how many frames within a slice must correspond to an identifier before the entire slice can be labelled as such. This requires introducing a threshold and using extended contextual features. For example, a slice containing only two identifier frames should not be labelled as an identifier, even though it partially overlaps with one. In our research work, if the slice contains 80% of an identifier, the whole slice is considered as the identifier.

In summary, while equal slicing allows for transcription-free classification, it also introduces several limitations that motivate the exploration of alternative input-slicing strategies.

Figure 4.5: Frame and Equal Slicing



Finally, the window input used in this research is frame-wise. A frame represents a very small segment of audio, and in our experiments, we use frames of 0.25 ms with a 0.20 ms hop, resulting in 0.05 ms of overlap. An illustration of this input size is shown in Figure 4.5. The frame-wise slicing

strategy can be seen as a compromise between the word-level and equal-slicing approaches, combining several of their advantages.

First, similar to equal slicing, the frame-wise method does not rely on transcription, which is essential for our speech anonymisation task. Second, it avoids the problem encountered in the word-slicing approach, where words may be split into two unequal parts. In contrast to equal slicing, it also eliminates the need for additional thresholds during training, because each frame can be precisely aligned with the word boundaries, ensuring reliable labels.

However, these benefits come with important trade-offs. The primary disadvantage is that the resulting sequence lengths become extremely large, which significantly increases computational requirements and training time. Moreover, frame-level features tend to be noisy and often carry little or no semantic information. Still, using this representation allows us to investigate whether speaker identifiers can be predicted purely from acoustic characteristics.

Another challenge is that frame-wise predictions require careful smoothing and pooling to produce interpretable results. As mentioned in (Tamir et al., 2025) the importance of contextual information and temporal smoothing in frame-wise audio predictions is frequently emphasised to improve robustness and produce interpretable outputs in speech and audio analysis. Post-processing may also be necessary: individual frames may be mistakenly classified as identifiers even when their neighbouring frames are not. To obtain precise predictions, the model must incorporate this contextual information - for example, by considering whether surrounding frames are also classified as identifiers.

All advantages and disadvantages of each input size are listed in Table 4.3.

Table 4.3: Comparison of input sizes for PII detection models

Input Size	Description (Advantages and Shortcomings)
Phrase-level	Advantages: captures broader semantic context; robust to noise in individual words; aligns with human perception. Shortcomings: low temporal precision; risk of over-redaction (whole phrase marked); variable length complicates training.
Word-level	Advantages: directly aligns with typical PII labels; balanced resolution (semantic + temporal); results are easy to interpret and evaluate. Shortcomings: relies on ASR/forced alignment; ignores sub-word cues; limited cross-word context.
0.5-second slicing	Advantages: uniform input length (easy batching); independent of ASR alignment; captures sub-word acoustic details. Shortcomings: ambiguous mapping to words/phrases; requires smoothing/aggregation; redundant due to overlap.
Framewise	Advantages: maximum temporal resolution; no reliance on segmentation; enables precise redaction boundaries. Shortcomings: extremely long sequences (computationally costly); very noisy at the frame level; hard to interpret.

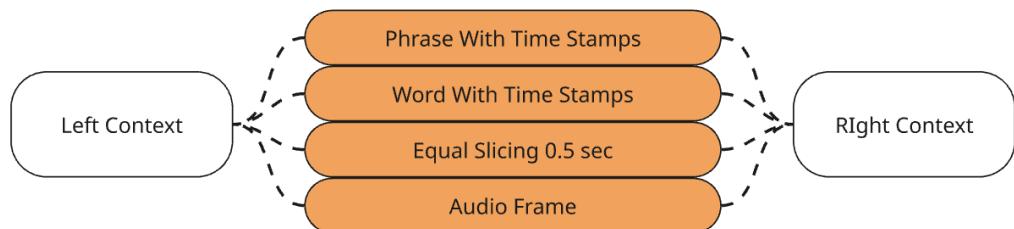
To sum up, the choice of input size is one of the key aspects of this work since different types of input size give different insights about the task of PII classification. Meanwhile, Phrase and word-level approaches give us clear interpretability and are more aligned with human language, but these two approaches lack independence from transcription. On the other hand, frame-wise and equal slicing approaches are more independent but less interpretable. Frame input should be one of the most important input sizes since it lacks the disadvantages of the input sizes and allows building a truly independent anonymisation system, but the size itself could be a problem since it is not certain that the model will be able to recognise anything within this small input. However, it will be interesting to see the results and analyse each of these approaches.

4.4.3 Context

Another question which will be answered in this work is how important context is in the PII classification task. Each of the implementations will have both no context and context versions, so by comparing the results with some confidence, we will be able to say whether context is crucial or not.

Figure 4.6 shows 4 different kinds of input and how they are fed to the model with context.

Figure 4.6: Model input



When the model is trained with context, it incorporates an additional 0.5-second slice of audio both before and after the primary input segment. This duration was selected because it is likely to capture the words before and after the target segment, which can be crucial for accurate classification. This choice is supported by the data shown in Table 4.1, where the average word duration in our dataset is approximately **0.32 seconds**. Therefore, including 0.5 seconds of contextual audio generally allows the model to have one word on each side of the input, potentially improving its ability to understand the surrounding context.

Experimenting with context input is specifically interesting in cases with frame-wise classification. Comparing frame-wise results with and without ± 0.5 context will show us how important external context is for classification, or whether it can be done without the external context, and the model can rely only on audio features of a particular frame and not on the whole context. This could provide valuable insight and pose an important objection for future research.

We think that the context for phrase-level will probably not be as useful as for smaller sizes, but sometimes it could be helpful. In general, we think that context and no context approaches in this case should demonstrate relatively the same results.

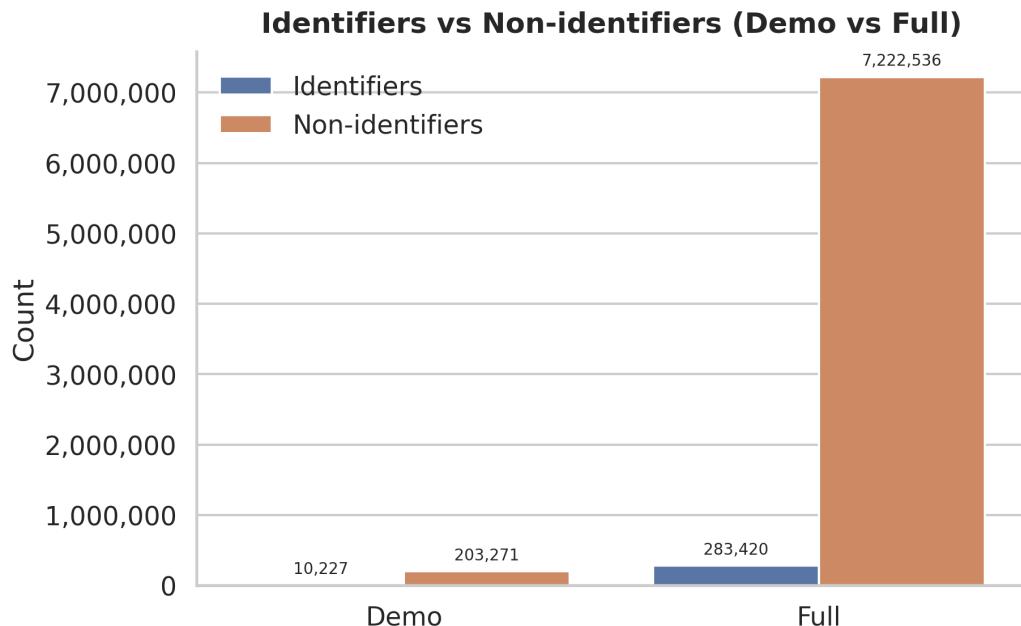
In the case of 0.5 equal-slicing-input, two additional contexts should be a solution to cases when words are split into two separate parts so the model will have a full representation of the word.

4.4.4 Binary classification setup and data distribution

Models will be used in two setups: multilabel and binary. Two configurations will help to determine how well the model generalises. Binary setup will probably be less sensitive to class imbalance; however, it will lack interpretability, whereas a multilabel setup will allow for determining model weaknesses and point out exact advantages and disadvantages of the model.

Figure 4.7 shows the distribution of identifiers and non-identifiers in datasets for demo training and the dataset which will be used for training with the full amount of data.

Figure 4.7: Distribution Figure



From the bar plot above, we can see that the demo training dataset consists of 213,488 words, and 10,227 of them are PIIs. For full dataset, the distribution of the labels is almost the same percentage-wise, with 7,222,536 non-identifiers and 283,420.

The distribution appears quite unbalanced, and the dataset is dominated by non-identifier classes. Percentage-wise, both datasets demo and full have the same distribution as, for example, in (Cohn et al., 2019b), where the Authors mention that their distribution in **SWFI Dataset and AMC'17 Medical Conversations** has the same unbalanced nature and identifiers are also only 5% of the whole dataset. Another work in the field of anonymisation, which mentions the unbalanced nature of data for the PII classification task in text domain, is (Ji et al., 2025). Relatively the same to ours label distribution for PII and NoPII labels in these two papers confirm that our dataset is sane and suitable for the task of PII classification, however, both of the paper highlights, the importance of evaluation metrics which are extremely important within setups where data is heavily imbalanced in order to have valuable results.(more details about evaluation metrics both for multilabel and binary setups provided in Section 4.6)

Binary classification setup for PII within the audio-only domain is a simple and reliable way to determine whether chosen models, are capable of seeing the difference between non-PII and PII. The binary classification setup will provide general knowledge of whether it is possible and to what extent to classify identifiers based only on audio feature input.

Binary setup will also provide valuable insights about the effect of different pooling techniques and help to determine optimal pooling technique for different models and input sizes.

Besides testing binary setup on several input sizes and different pooling technique another dimension which will be added both to binary and multilabel approaches is a context features.(More information about context and how it is added to the input can be found in Section 4.4.3)

Different models, pooling techniques, input sizes, and input context all together will help to create a broad and representative picture of binary PII classification task with audio features input.

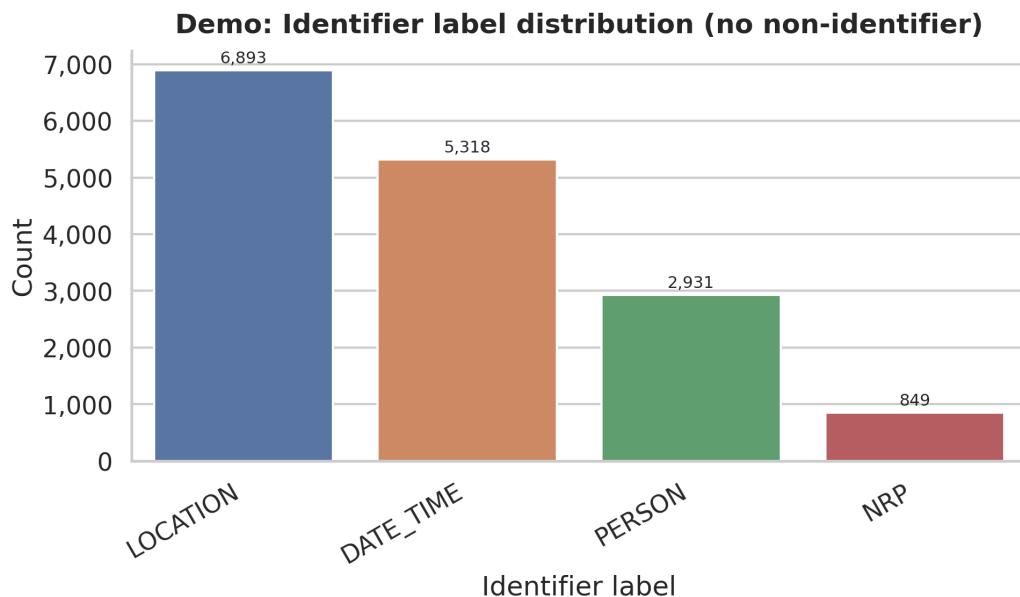
4.4.5 Multilabel classification setup and data distribution

Another setup in this work is multilabel classification with the same data. Multilabel approach will examine models' ability to distinguish not only identifiers from non-identifiers but also to determine which classes are more difficult to predict and which are easier.

Figure 4.8 shows the distribution of labels inside the demo dataset, which was used for binary classification as well. The non-identifier class was

removed to make the bar plot more informative. Since it is by far the largest class - 203,271 words in the training data - it would overshadow the other categories and make their bars difficult to see.

Figure 4.8: Filtered Distribution



In the plot above, we can see four classes. The most frequent class label represented in our data is LOCATION, followed by DATE_TIME, PERSON and NRP.

The distribution seems to be very unequal, and some of the classes have very few instances. However, pooling techniques should resolve the class imbalance issue and allow us to receive valuable results even in conditions where some of the labels are underrepresented.

A multilabel classification approach will also allow us to discuss the model's potential to distinguish between different PII classes. It also will help to find out which classes are more easily predicted and with which model struggles the most.

In general, this setup only differs from binary in the way that it utilises several classes. In all other aspects we will apply the same 4 input sizes same 3 different pooling techniques and the same 3 models that we have discussed in the Models 4.7 section. These two, Multilabel and Binary, setups will provide us reliable results which will help to determine models behaviour and its struggles in the task of PII classification within audio-feature only domain.

4.5 Optimisation, hyperparameters and Pooling Techniques

Once all preparation with datasets is done, it is time to discuss the exact Pipeline of training. In the following section Pooling Techniques, Loss Function and Evaluation, metrics will be discussed.

Figure 4.9: Training pipeline schema.

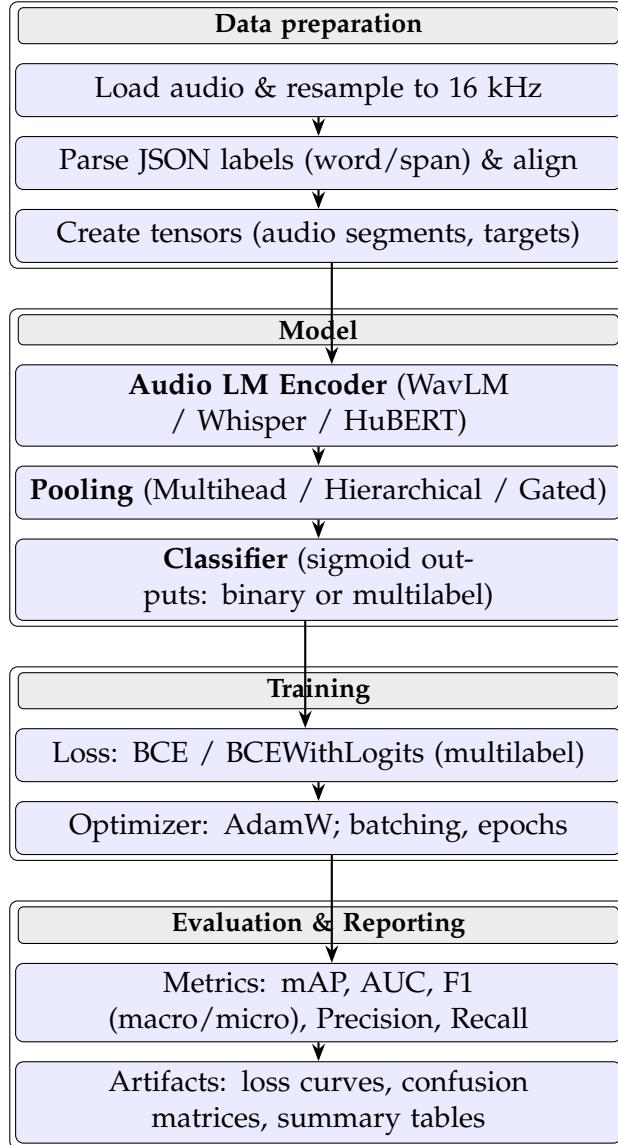


Figure 4.9 shows the general schema for the training process.

First step is to take audio together with label, and then resample the

audio to 16kHz, just in case we missed some of the audio in the Data Pre-processing step.

Resampled audio, then converted into a tensor, which was further encoded with the model's encoder. To encode piece pooling technique is applied, and the model classifies the input. During training, BCE loss was used both in multilabel(with logits) and binary approaches and then weights were optimised with the AdamW optimiser. Both BCE loss and AdamW mentioned as a solid approach to treat extremely unbalanced data during the training (Pang et al., 2022)

Models were trained on 10 epochs in the case of the multilabel approach and on 5 in the case of the binary. The number of epochs was determined empirically.

As an output, after training, results are saved with metrics, a confusion matrix and loss curves plots. The results will provide valuable insights about models' performance both for binary and multilabel classification approaches.

4.5.1 Pooling Techniques

As mentioned at the beginning of this section, this research work employed three different pooling techniques. The pooling technique is a crucial parameter in this work since it helps to reduce high-dimensional sequences to a compact representation, which helps the model to focus on the most relevant information and improve efficiency and accuracy. Also, choosing a pooling technique is important because the dataset for the training task is unbalanced and requires proper treatment of an unbalanced number of classes. If we want to achieve valuable results, it is crucial to choose and experiment with different pooling techniques to figure out which works best. Moreover, in this research, pooling techniques will be tried out both on multilabel approaches and binary ones, which will allow us to understand which pooling technique is better for different approaches in terms of distribution.

Since we are working with computationally intensive audio data, it is essential to have as much information as possible with minimal memory utilisation. The pooling technique is a crucial tool that can help resolve this problem. Pooling applied to our data will condense temporal information into a meaningful fixed-size vector.

In this section, three chosen pooling techniques will be discussed in detail, namely: multi-head attention pooling, hierarchical pooling, and gated

pooling. Each of these approaches can help manage memory usage and, as highlighted in recent work on audio emotion recognition(Leygue et al., 2025), attention-based and hierarchical pooling methods are crucial for localizing high-value information within sequential audio signals. This is particularly relevant for speech anonymisation, where the model must reliably distinguish important PII from silence and other non-relevant words.

The first pooling method applied in this work is gated pooling. Here, the input to the pooling layer is a sequence of frame-level embeddings $\{x_i\}_{i=1}^T$, where $x_i \in \mathbb{R}^d$ denotes the d -dimensional representation of the i -th audio frame produced by the encoder (WavLM, Whisper or HuBERT), and T is the number of frames in the segment. Gated pooling introduces a learnable gate for each frame that decides how much this frame should contribute to the final utterance-level representation. The gating operation is defined in 4.1.

$$\begin{aligned} g_i &= \sigma(Wx_i + b), \\ h &= \sum_{i=1}^T g_i x_i, \end{aligned} \tag{4.1}$$

where $W \in \mathbb{R}^{1 \times d}$ and $b \in \mathbb{R}$ are learnable parameters, $\sigma(\cdot)$ is the sigmoid activation function, $g_i \in (0, 1)$ is the scalar gate associated with frame i , and $h \in \mathbb{R}^d$ is the pooled representation of the whole audio segment. Intuitively, frames that are predicted to contain informative content (potential identifiers) receive gates g_i close to 1, while silence or background frames are assigned gates closer to 0, thereby reducing their influence on h .

A more expressive pooling technique used in this thesis is **multihead attention** pooling. In the single-head case, attention assigns a normalised weight a_i to each frame based on its content and the global context of the segment:

$$a_i = \frac{\exp(\mathbf{w}^\top \tanh(\mathbf{V}x_i + \mathbf{b}))}{\sum_{j=1}^T \exp(\mathbf{w}^\top \tanh(\mathbf{V}x_j + \mathbf{b}))}, \tag{4.2}$$

$$h = \sum_{i=1}^T a_i x_i, \tag{4.3}$$

where $\mathbf{V} \in \mathbb{R}^{k \times d}$ and $\mathbf{b} \in \mathbb{R}^k$ are learnable parameters of a small feed-forward layer, $\mathbf{w} \in \mathbb{R}^k$ is a learnable vector projecting the non-linear

transformation to a scalar score, and a_i are attention weights satisfying $\sum_{i=1}^T a_i = 1$. In the multihead variant used in our experiments, this mechanism is applied in parallel for several heads $h = 1, \dots, H$, each with its own parameters, and the resulting head-wise pooled vectors are concatenated and fed to the classifier.

Compared to gated pooling in Equation 4.1, attention pooling (Equations 4.2–4.3) does not weight each frame independently. The normalisation across all frames means that the weight assigned to a given audio frame depends on the scores of all other frames in the segment. This allows the model to use the context of the entire utterance when deciding which frames (and therefore which words or subword pieces) are most relevant for predicting the presence of personal identifiers.

One more advantage of using attention pooling for the task of PII classification is that it could be easily extended from binary classification to multilabel classification. This particular architecture of attention mechanism makes this pooling technique the main candidate to become the best approach for multilabel classification since it is not only learning label-wise attention but also the context of each label.

Capturing external context maybe crucial, since in some of examples bellow, the exact context could may be help to determine whether the next or previous word is an identifier or not.

1. My name is **PERSON**
2. I was born in **DATETIME**
3. I am **DATETIME** years old
4. I am from **LOCATION**
5. I believe in **NPR**

In the examples above, we can clearly see that in some particular cases, the surroundings of identifiers are more or less common, especially in cases when a person introduces themselves. A model can also learn these patterns and base its decisions regarding the target frame mainly on the surroundings and not on the features of the identifiers themselves. To find out what is more important, the context or the target feature itself, another pooling technique is required.

Hierarchical pooling is the last technique that will be used, which could be a better choice than the other. Unlike the previous pooling approach,

hierarchical pooling does not treat all audio inputs the same way and has several levels of input window(macro and micro levels), which will guarantee that the model will at least understand the difference between silence and words and capture the context of each input.

In this work, hierarchical pooling is applied on top of the frame-level encoder representations. The sequence of audio frames is first divided into a series of short, overlapping *coarse windows*. For each window, multihead attention is used to compute a local summary vector that aggregates the most informative frames within that window, while down-weighting silence and background noise. In a second stage, another multihead attention layer operates over the sequence of window summaries to produce a single phrase, word, slice, frame representation. This two-stage design allows attention to be distributed not only over individual frames but also over clusters of frames, improving a multiscale representation that captures both local details and broader temporal context.

Concretely, we use hierarchical pooling with multihead attention at the *micro* level (within windows) and at the *macro* level (across windows). The micro level helps distinguish silent or non-informative frames from those that are likely to contain PII, while the macro level models longer-range relations between words and segments within a phrase. Similar ideas have been explored in speech emotion recognition, where hierarchical pooling has been shown to suppress irrelevant frames and emphasise emotionally salient regions (Leygue et al., 2025). By capturing both local and global patterns in this way, the hierarchical pooling head is well suited for our binary and multilabel identifier classification tasks.

4.6 Evaluation Metrics

Another cornerstone of this research work is the calculation of the final output of the model. Since in this research two classification strategies are used: binary(PII and No PII) and multilabel (LOCATION, DATE_TIME, PERSON, NRP), each of them requires its own way of evaluation and a different set of metrics.

It was decided to abandon the traditional accuracy metric in favour of more representative ones. Although, Accuracy could be a good metric in some of the tasks it does not suits for our particular task. Accuracy is misleading in this context because our datasets are highly imbalanced. It cannot distinguish genuine performance from simple overfitting - for example, a model that predicts NoPII for nearly every instance would still achieve high accuracy.

Table 4.4 lists the metrics which were chosen for binary and multilabel classification.

For binary classification, the F1 score can provide a more reliable answer on whether the model classification is sane and is not overfitting. To calculate the F1 score, Precision(high=fewer false alarms) and Recall(high=better identify sensitive information) are used, which separately give information about each tag and highlight how the model is successful with detecting PII and non-PII. Together, these metrics effectively evaluate the overall quality of predictions by accounting for both missed detections and false positives, including the influence of the dominant (non-PII) class.

Table 4.4: Compact summary of evaluation metrics

Binary metrics	Multilabel metrics
	mAP = $\frac{1}{K} \sum_{k=1}^K AP_k,$
Precision = $\frac{TP}{TP+FP}$	AUC (macro) = $\frac{1}{K} \sum_{k=1}^K \int_0^1 TPR_k(f) df$
Recall = $\frac{TP}{TP+FN}$	AUC (micro) = $\int_0^1 \frac{\sum_k TPR_k(f)}{K} df$
$F_1 = \frac{2PR}{P+R}$	F1 (macro) = $\frac{1}{K} \sum_{k=1}^K \frac{2P_k R_k}{P_k + R_k}$
	F1 (micro) = $\frac{2 \sum_k TP_k}{2 \sum_k TP_k + \sum_k (FP_k + FN_k)}$

For multilabel classification, another set of complementary metrics was chosen another which will provide more or less the same insights on the model's output for a multilabel setup. To have a wider picture, five particular metrics were chosen, namely: Mean Average Precision, Average Precision, AUC macro and micro, and F1 macro and micro.

- **mAP:** this metric will show how well the model ranks positive examples ahead of negative ones across all classes by averaging AP_k (AP-average precision, k-class) for each class.
- **Macro-AUC:** shows how well the model distinguishes classes from each other by averaging the sum of True Positive Ratio ($TPR_k(f)$) of each class.
- **Micro-AUC:** shows how the model ranks corrects over wrong ones by weighting each instance equally.
- **F1 micro:** how well the model predicts frequent labels, by using global sums of true positives TP_k (True Positives), false positives FP_k , and false negatives FN_k (False Negative) across all classes

- **F1 macro:** how well the model predicts overall, by averaging F1 score over the classes.

These metrics should be enough to have a good overview of the model’s performance, and they will allow us to explain the results and dependencies between model choice, context influence, and pooling techniques.

In this work, for training with multiple classes, we evaluate our models using a multilabel classification framework rather than a traditional multiclass setup, as certain input sizes, such as phrases or potentially equal slices, may include multiple labels simultaneously. For Phrase-level, we treat a prediction as correct only when all labels within the segment are predicted correctly, because this input type may contain multiple classes simultaneously. An additional safeguard is that the NONIDENTIFIER(NoPII) class can appear only on its own, never alongside PII classes.

This strict evaluation is less meaningful for word-level and frame-wise inputs, where each instance typically corresponds to a single label. However, we apply the same procedure to maintain consistency across all reported metrics and to avoid introducing additional evaluation complexity. It is important to keep in mind that the results for Word and Frame-wise inputs could potentially be improved by using a multiclass (instead of multilabel) formulation specifically for these two input sizes.

4.7 Models

The choice of the classification model is one of the most important parts of this work, since when analysing the results we will rely on the model’s output. Therefore, it is crucial to choose the right model which will not only be capable of processing audio input but also effectively extract potentially important audio-features such as prosody, tonal features, intonation and anything else which could help to classify audio as a PII or not.

The Transformer architecture has become a cornerstone of modern machine learning, with successful applications in audio-related tasks such as STT, TTS, and ASR. Since the main goal of the work is largely reduced to the classification of audio segments, it is first worth paying attention to pre-trained models for audio classification.

In the sections below, there will be a detailed overview of the three models that were chosen for the audio classification task, examining their

architecture in detail, and what potential insights about identifiers they can provide based on the architectural structure of each model.

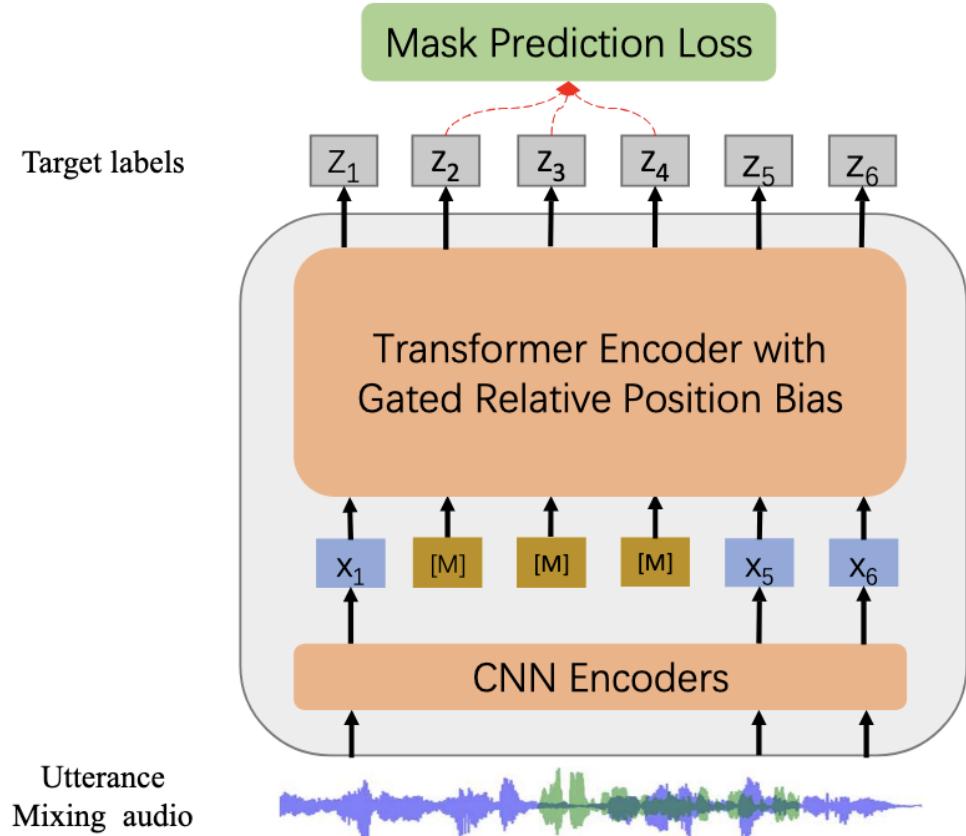
Also, it is worth mentioning that all of the models for training were relatively the same size parameter-wise for more precise and fair comparison(95 mil - WavLM/Hubert; 74 mil - Whisper)

4.7.1 WavLM model

The WavLM model was introduced at the end of 2021 (Chen et al., 2022) and right away became one of the leading models for the self-supervised speech representation task. The WavLM model was designed as a tool that is suitable for various full-stack speech processing tasks, including automatic speech recognition(ASR), speaker verification, diarization, and many other paralinguistic applications.

Figure 4.10 shows the architecture of the WavLM model. The first block of the model is a stack of seven temporal convolutional blocks which process raw audio waveforms and transform it into a frame-level features representation. Encoded features are about 25 ms long and contain information about both acoustic and linguistic properties. Then the features from the convolutional block are passed to the Transformer encoder, where the self-attention mechanism is augmented with a gated relative position bias.

Figure 4.10: Figure from WavLM paper (Chen et al., 2022)



The innovation in the WavLM model lies in the attention mechanism and how the model treats the encoded input. Gated Relative Position Bias allows the model not only to pay attention to each part of the audio but also to the relative position of other frames. Model architecture also allows us to adjust how much we care about the relative positions of two particular sounds.

A recent study (Miara et al., 2024) successfully used the WavLM model for speaker identification by extracting voice features. Their approach fine-tunes the pre-trained WavLM using a self-supervised learning framework combined with supervised pseudo-labels, allowing the model to capture speaker-specific representations effectively. This work demonstrates WavLM's versatility beyond its original speech recognition tasks, achieving competitive results in speaker verification through a refined embedding extraction and attention-based back-end.

Other work that uses Wavlm architecture is the most recent paper (Sun et al., 2025), where authors use the WavLM model as a feature extractor

for overlapping speech detection. As an output, researchers managed to get meaningful results and outperform prior methods such as: CNN, x-vectors, pyannote, and XLSR-Conformer. Also, their results outperform baseline by 3 points(ours' vs XLSR-Conformer: F1 82.76% vs. 79.21%).

Other work that proves that WavLM is the leading model in the task of speaker diarisation is (Song et al., 2025). In this work authors also managed to successfully extract frame-wise feature representation for further diarisation tasks. In terms of metrics, according to Diarisation Error Rate(DER),the researcher's approach takes 3rd place in the AVSD track of the Multimodal Information-Based Speech Processing Challenge (MISP).

Several studies highlight WavLM's effectiveness as a feature extraction model, particularly for speaker diarization applications. Using the Gated Relative Position Bias architecture, researchers managed to get meaningful results. This particular feature (Gated Relative Position Bias) of the model could be extremely important for identifier detection and play a key role in the task of anonymisation. Some PII is not expressed as a single word but rather as a specific sequence of words or phrases that, when spoken in a particular way, can reveal clues about a person's identity. A gated approach may help detect such sequences: if the model can reliably identify certain frames as containing identifiers, we can infer - with some degree of confidence - that these words or phrases carry specific audio characteristics that the model can capture.

Additionally, previous research in speaker diarisation shows that features extracted using the WavLM architecture can effectively encode speaker-related information. This suggests that WavLM-based representations could also be leveraged to uncover relationships between different types of identifiers.

Thanks to its architecture, the WavLM is probably one of the most relevant models for speech anonymisation and for the PII classification task in particular. Its ability to capture not only acoustic properties but linguistic as well makes it a powerful tool for this research. Several research works and benchmarks confirm that the WavLM model is still a state-of-the-art solution in the task of self-supervised speech representation. Moreover, comparing the performance of this model to HUBERT allows us to see if Gated Relative Position Bias is an effective architecture in the task of speech anonymisation (Hubert description will be provided in the following section). We think that the WavLM model should demonstrate the best performance among all models and provide better results due to its architecture and task-oriented structure.

4.7.2 Hubert model

The HUBERT model was introduced in 2021 (Hsu et al., 2021). HUBERT model utilises the same BERT-like masked prediction strategies but applies them to audio data input. This BERT-like architecture allows the model to generate powerful, generalizable representations directly from raw waveforms.

HUBERT architecture is relatively similar to the WavLM model. It receives a raw audio waveform and processes it through a stack of seven convolutional blocks. The main block of the model is inspired by the BERT architecture (Devlin et al., 2019), which uses multilayered transformers with long-range dependencies and context within an audio sequence.

Unlike other models, which learn from contrastive and reconstruction tasks, the Hubert model masks audio chunks and predicts targets; this particular strategy allows the model to learn different speech patterns, speakers' identity and linguistic structures as well. This architectural idea allowed the authors of the model to outperform the previous state-of-the-art model, wav2vec, in many different tasks. Moreover, Hubert turns frames 25ms into powerful features which represent different parts of speech and will be extremely useful in the task of speech anonymisation

Figure 4.11: Figure from original Hubert paper (Hsu et al., 2021)

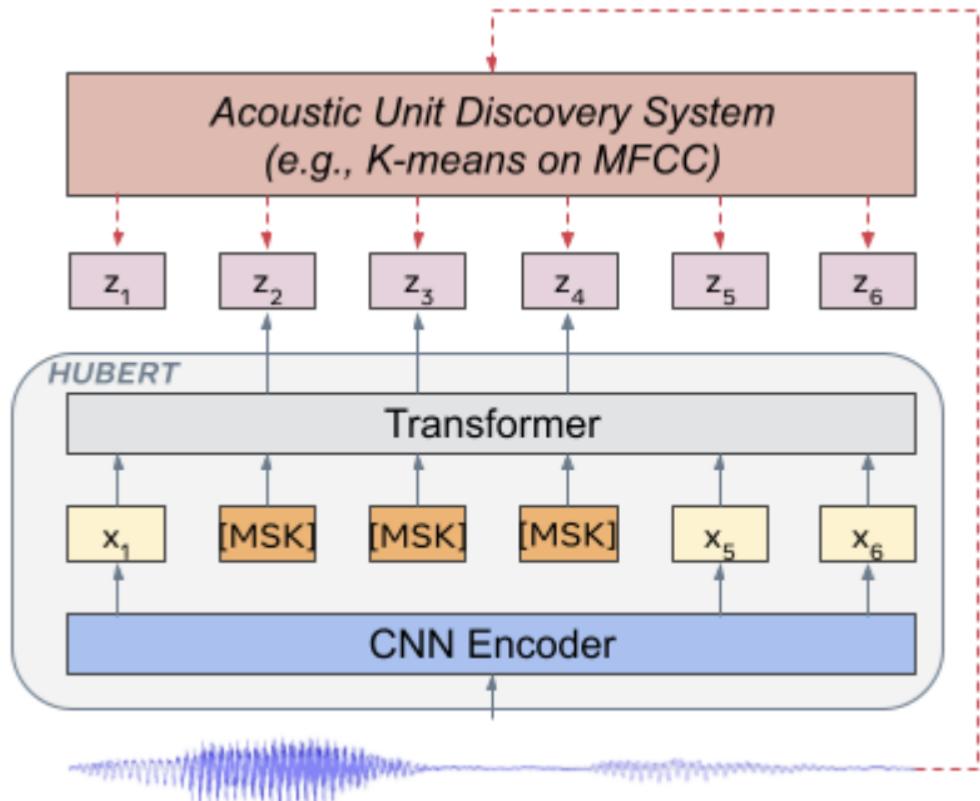


Figure 4.11 shows more precisely how the model handles the input and which architecture and layers it has.

After release, HUBERT became a state-of-the-art model replacing the wav2vec architecture, which had been previously considered the best model for speech self-supervised representation learning. The authors managed to adapt the BERT architecture, which is supposed to work with text, to continuous audio data. The model learns to predict masked segments using iterative K-means clustering. This algorithm of prediction could be effective in the task of speech anonymisation. Moreover, one of the papers (Miao et al., 2022) mentions that middle and higher layers capture rich linguistic content, meanwhile lower layers preserve speaker identity.

Hubert's approach is also well known in the task of voice anonymisation. In (Yao et al., 2024), authors use the HUBERT model to perform the voice anonymisation task. In the conclusion authors mention that speech features can be separated into identity-related features and content-related components. We can use this insight as a hint that this model could also

perform not only voice anonymisation but also speech PII anonymisation tasks.

4.7.3 Whisper model

OpenAI introduced the Whisper model in September of 2022⁵. The model was designed for multilingual transcription and translation tasks. Its main strength is the amount and diversity of training data(680.000 hours). The amount of training data in particular makes the performance of Whisper sustainable and consistent even for audio recordings with different accents, background noise and noisy data.

Figure 4.12: Figure from original Whisper paper (Radford et al., 2022)

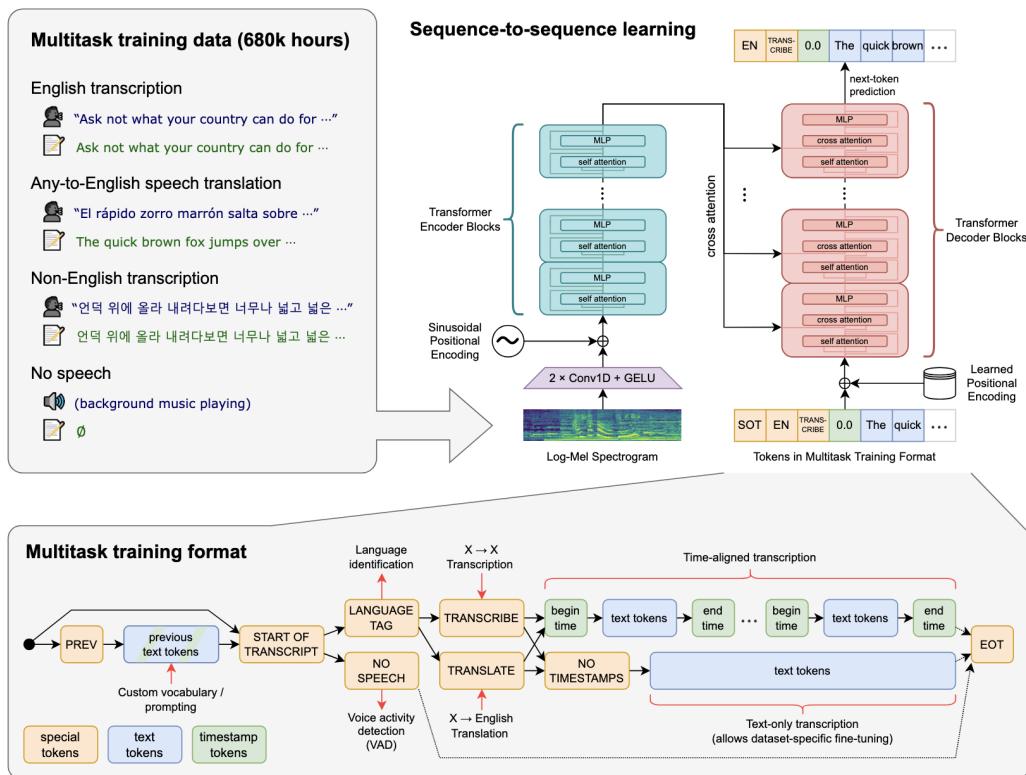


Figure 4.12 shows Whisper's architecture and how the model processes audio data. From the figure, we can see that the Whisper model from OpenAI has a Transformer encoder-decoder architecture. The model splits audio input into 30-second segments, audio then converts to a log-Mel

⁵ <https://openai.com/research/whisper>

spectrogram before encoding and decoding. After the model receives a spectrogram, it transcribes speech, detects its language, and performs voice activity detection.

Although Whisper was initially developed as a speech recognition system and trained to perform speech-related tasks, such as speech recognition, translation, and voice activity detection, it could be used for other purposes as well. It can also perform time stamp generation based on the log-Mel spectrogram; this feature of Whisper was used during the preprocessing stage in our research work.

Whisper is a multipurpose model indeed, and its architecture allows it to perform different tasks and classification tasks are one of them. One of the papers where we can see successful usage of Whisper as a classification model is (Ma et al., 2024). In this paper, the author uses Whisper as an audio classification model with a zero-shot approach. The author claims that without any finetuning and only with the help of prompt-based techniques, it managed to outperform the baseline by 9% on average over multiple test datasets.

Another paper also showcases that the Whisper model could be used as a slot information extractor(Li et al., 2024). This study highlights that Whisper has deep audio content understanding and effectiveness in the task of classification, showing also that Whisper is able to classify intents.

Whisper's ability to classify and its broad architecture, which allows the capture of different audio features based on log-Mel spectrograms, could be useful in the task of PII classification. Moreover, it is interesting to see how this model will perform for the anonymisation task, keeping in mind that it was also used as a tool of transcription in this research.

Chapter 5

Results

In this research work, 144 experiments were set up in total. All of the training processes were evaluated as described in Section 4.6. In the following Chapter, results for the Binary and Multilabel approaches will be presented, and the best solutions for each input level will be stated. Also, it is worth mentioning that not all results will be provided; mainly, we will take a look at the results of training with context input and sometimes refer to Appendix 6.2 for comparison with no context training instances in cases where context turned out to be crucial for results.

Results will be compared to each other, primarily relying on three major metrics within a multilabel setup: F1 micro, F1 macro, and mAP; in the binary setup, models will be compared to each other with the F1 metric. Further discussion regarding models' performance hypotheses and conclusions will be provided in 5.7 in this section; models' performance will be compared to each other, and the best approaches for each input size will be named.

5.1 Multilabel results

In this section, results for Multilabel approaches will be examined to determine the best model and pooling technique in the task of speech PII classification. At the end of this section, the best combination of pooling technique and model for each input size will be stated.

5.1.1 Multilabel Phrase input window

In the Table 5.1, results for Phrase input with context can be observed. The results show that both pooling techniques and model selection are crucial in the PII audio classification task. Some of the models rely more on pooling techniques, some less.

Table 5.1: Phrase-level multilabel PII classification results for different models and pooling strategies.

Model	Pooling Strategy	mAP	AUC (macro)	AUC (micro)	F1 (macro)	F1 (micro)
WavLM		0.3285	0.7990	0.8297	0.3486	0.3813
HubERT	Multihead-Attention	0.1719	0.6768	0.7045	0.1884	0.2458
Whisper		0.4480	0.7712	0.9135	0.4451	0.7677
WavLM		0.3638	0.8160	0.8495	0.3784	0.4576
HubERT	Hierarchical Pooling	0.2041	0.7012	0.7534	0.2199	0.2679
Whisper		0.2552	0.7630	0.8813	0.2844	0.6155
WavLM		0.3398	0.8090	0.8494	0.3044	0.3841
HubERT	Gated Pooling	0.1862	0.6728	0.7517	0.2107	0.2299
Whisper		0.2356	0.5682	0.5579	0.2485	0.5228

Among 9 different training instances, we can see that the Whisper-Multihead attention approach clearly outperforms every other in every parameter except AUC (macro), which is slightly better with WavLM and multihead attention domain,n and with hierarchical pooling, WavLM is better in general in terms of macro AUC. However, despite the best overall performance, we can also see that it really depends on the pooling approach and performs better with the Multihead Attention approach, on average 30 per cent worse with the Gated one.

Hubert, meanwhile, became the worst model in the multilabel setup with the Phrase input window size, among others, in every setup, regardless of the pooling approach.

WavLM is more stable and consistent in terms of metrics and more or less performs the same with every pooling technique.

Appendix 6.2 has information about models' performance results without context, and from there, we can see that WavLM and Hubert models perform slightly better in terms of mAP F1 micro and macro scores without additional context, with multihead attention pooling; meanwhile, additional context to the Whisper model with multihead attention slightly improves results within same metrics. Within Hierarchical Pooling WavLM and Hubert have the same trend as in multihead attention pooling, but the differences are also not that noticeable. Whisper with

hierarchical pooling has better results among the three major metrics but significantly differs only in one F1 micro with context equal to **0.61**, meanwhile, without context, only **0.46**. With a gated pooling setup, WavLM also has almost the same results for mAP, F1 micro and macro and performance is slightly better without context. Hubert model, meanwhile, has a dramatic improvement in terms of F1 micro with a 0.81 value within the no context approach and 0.22 with context. Whispers results in gated pooling are almost identical, with a very small advantage to the no-context approach.

In general, all of the models demonstrate meaningful results and the ability to learn some relations between PII and its embeddings. Also, results for phrase level look consistent and logical, keeping in mind the architecture of the models, the amount and type of data they were trained and the task. As expected, the best approach for a broader input window is Whisper with multihead attention and the worst performance for Phrase input size is demonstrated by the less advanced model Hubert, and it is consistently worse than any other model+pooling combination. Context appeared to be noticeably important only for Whisper approaches.

5.1.2 Multilabel Word input window

Another input size explored in this research is the word-level input window. Unlike the phrase-wise input size, this one is not that useful because it is architecturally not applicable to a real-world task. However, the model's results can be useful in terms of interpretation in further discussion and can help to determine whether the model looks at the whole phrase or only at a particular word.

Training results for word-level input are shown in Table 5.2. As well as in previous input size, word-level input also relies on both the pooling technique and model selection, but the output results differ even more depending on the pooling approach.

Table 5.2: Word-level multilabel PII classification results for different models and pooling strategies.

Model	Pooling Strategy	mAP	AUC (macro)	AUC (micro)	F1 (macro)	F1 (micro)
WavLM	Multihead-Attention	0.3060	0.8645	0.9739	0.3577	0.9362
		0.2692	0.8023	0.9545	0.3059	0.9124
		0.4969	0.9864	0.9851	0.4814	0.5290
HuBERT	Hierarchical Pooling	0.3695	0.9184	0.9381	0.3721	0.4381
		0.1137	0.5708	0.6053	0.1295	0.1478
		0.2045	0.5148	0.7615	0.0932	0.1823
Whisper	Gated Pooling	0.2426	0.9152	0.9311	0.2912	0.3450
		0.0907	0.8009	0.8093	0.1275	0.1285
		0.2014	0.5085	0.6949	0.2132	0.8647

From the table, we can see that Whisper is again becoming a leader among other approaches, as well as in the Phrase input window scenario. However, in comparison to phrase-level, Whisper performs better than other models only with multihead attention pooling.

WavLM, meanwhile, again demonstrates more or less stable performance and is comparable to the phrase level in terms of metrics. With multihead attention pooling, it even managed to improve the performance in terms of F1 micro by a factor of 3, which means that for the model, it is far easier to find true positives on the word level than on the phrase level. Also, we can see that WavLM outperforms Whisper with Hierarchical and Gated Pooling approaches by far.

Hubert model has improved its performance in terms of F1 micro metric as well, but only in the case of the multihead attention approach. The rest of the metrics signal that the general performance of Hubert remains the worst among other models, regardless of the pooling technique.

Context within word level in multilabel classification setup appeared to be a detrimental for WavLM model with multihead attention pooling, and results are better without any context; with other pooling techniques, models perform almost the same both in no-context and context setups, but still without context slightly better across all metrics. Hubert meanwhile, seems to be very dependent on the context with word input size and multihead attention pooling since F1 macro = 0.19, F1 micro = 0.20 and mAP = 0.22 this is noticeably worse then with context; within hierarchical and gated pooling there is almost no difference in results between context and no-context setups (very slightly better with context in hierarchical pooling case and barely noticeably better in gated pooling without context). Whisper results with multihead attention pooling without context are way better than with context, with the following metrics values: F1 macro = 0.69, F1 micro = 0.65, mAP = 0.67. For

hierarchical pooling performance without context also appeared to be not that crucial as in the multihead attention case, but slightly better(same for gated pooling). Seems like in a multilabel setup for word classification, in most of cases, context appeared to be an obstacle for the model rather than help.

To summarise, models demonstrate decent performance and very consistent results. In comparison to the phrase level, we can see that despite Whisper remaining the leader among other Model+pooling combinations. However, we can see that the WavLM model, unlike Whisper and Hubert, still does not depend that much on pooling techniques and performs relatively the same. Meanwhile, Whisper and Hubert's performance noticeably degrade with hierarchical and gated pooling in comparison to the multi-head attention one.

5.1.3 Multilabel Equal input window

Table 5.3 shows the results for the equal slice approach. Unlike the previous two examples of input sizes, this one does not rely on any predefined timestamps, which makes it more applicable to a real case scenario when the model will not have any exact time stamps.

Table 5.3: Equal-slicing (0.5 s) multilabel PII classification results for different models and pooling strategies.

Model	Pooling Strategy	mAP	AUC (macro)	AUC (micro)	F1 (macro)	F1 (micro)
WavLM	Multihead-Attention	0.3060	0.8645	0.9739	0.3577	0.9362
		0.2692	0.8023	0.9545	0.3059	0.9124
		0.1934	0.5156	0.8645	0.2011	0.8514
HuBERT	Hierarchical Pooling	0.3459	0.8616	0.9840	0.3921	0.9444
		0.2469	0.7497	0.9732	0.2815	0.9268
		0.1941	0.5486	0.8285	0.2054	0.7399
Whisper	Gated Pooling	0.2971	0.8367	0.9383	0.3166	0.9032
		0.2236	0.7057	0.8815	0.2466	0.8789
		0.1917	0.5133	0.7725	0.2025	0.6860

From the results table, we can see that in the equal slice input approach best model title is overtaken from Whisper by the WavLM model. We can see that by all metrics, WavLM with hierarchical pooling noticeably outperform other setups. So far, results from other input sizes for the WavLM model seem to be the most consistent in terms of performance.

The Hubert model is not an outsider anymore and managed to outperform the Whisper model in every pooling strategy approach. Despite it

outperforming Whisper by far, it is still not good enough for WavLM and could not demonstrate better results in any of the metrics among all pooling techniques.

Whisper's equal slicing approach is not performing that well. In the table, we can see that in comparison to word-level Multihead attention, results significantly degrade by all of the metrics and only the F1 micro metric increased. Although we can see a significant drop in the performance in comparison to word-level input multihead attention approach, other pooling techniques turned out to work better with the equal slicing approach than with word-level. Improvements can be observed in the Hierarchical Pooling approach, almost among all metrics except mAP, but according to F1 scores, general performance within hierarchical Pooling for Whisper is way better than for the word level.

So far, model results for Hubert and WavLM show improvement with narrower input windows. The opposite occurs with the Whisper model, which shows consistent degradation of performance and has better results with wider input windows.

Context does not really improve or degrade the performance for WavLM, and the results within all pooling techniques are almost the same according to Appendix 6.2. The results differ within the range from 0.01 to 0.03 points among three major evaluation techniques for the multilabel approach. The Hubert model also does not really differ in terms of performance with and without context approaches. Whisper follows the trend of two other models and has almost identical results both for no-context and in-context setups.

It is worth remembering that the equal slicing approach is not perfect, since labels which are assigned to slices have a certain threshold of 0.8, which means that slices which has a lower percentage of identifiers are not marked as an identifier. This particular approach of equal slicing seems to be working one however, this disadvantage with label assignment could seriously ruin the possible capabilities of models to predict labels correctly. Next, a frame-wise approach was taken into account to resolve this issue with equal slicing shortcomings and also provide insights regarding models' abilities to classify really short pieces of input information.

5.1.4 Multilabel Frame input window

Frame-level input window resolves several issues which occur with other approaches: uncertainty of input slices caused by the threshold in the equal slice level approach, and the requirement of exact time stamps of

words and phrases. Frame level resolves all of these shortcomings of previous approaches and could possibly help to create a decent PII classification system which works independently of predefined timestamps and thresholds. However, in the case of frame-level approach, we potentially face a lack of context, which is useful in the task of PII classification.

Table 5.4 shows the results for the frame-level approach, and from first glance, the results seemed to be very interesting and unexpectedly good.

Table 5.4: Frame-wise multilabel PII classification results for different models and pooling strategies.

Model	Pooling Strategy	mAP	AUC (macro)	AUC (micro)	F1 (macro)	F1 (micro)
WavLM	Multihead-Attention	0.5629	0.9581	0.9815	0.5646	0.9450
		0.4590	0.9333	0.9488	0.4567	0.9369
		0.2018	0.5226	0.7476	0.2147	0.5435
HuBERT	Hierarchical Pooling	0.3199	0.8524	0.9464	0.3676	0.9349
		0.3336	0.8463	0.9628	0.3760	0.9375
		0.2026	0.5311	0.7350	0.2164	0.6743
Whisper	Gated Pooling	0.3212	0.7542	0.7926	0.3597	0.9429
		0.2745	0.7788	0.9109	0.3161	0.9322
		0.2022	0.5050	0.6159	0.2133	0.6565

From the table above, we can see that, as well as in the equal slice window approach, the best model for frame-wise classification turned out to be WavLM in combination with Multihead Attention. According to metrics, WavLM, together with multihead attention, completely outperforms every other approach in a frame-wise setup. Moreover, this particular set-up demonstrates not only the best performance among all on the frame level but in general in a multilabel scenario in terms of mAP, F1 macro, F1 micro, and AUC micro and macro.

Hubert, as well as WavLM, demonstrated strong performance in comparison to previous training processes with other input windows. Hubert's performance is relatively the same as WavLM's one but just slightly worse, and this claim is applicable among all pooling techniques. However, as well as WavLM, it outperforms Whisper by far and does it with any pooling technique. In comparison to previous equal slicing input, we can see that Hubert is improving performance in all pooling techniques, and especially in Hierarchical and Gated pooling approaches. In general, we can see that Hubert works better with smaller input sizes.

Whisper, meanwhile, has the worst results for frame input level among all models and pooling combinations. Apart from the fact that Whisper performs worse than Hubert and WavLM, it also shows that the pooling technique does not affect its performance, and different combinations

of pooling techniques and models perform relatively the same in all of the Whisper-frame setups. Whereas in earlier experiments, multi-head attention pooling showed a clear advantage, here all metrics remain roughly the same regardless of the pooling method used.

Hubert and WavLM perform the same with and without context among multihead attention and hierarchical pooling techniques(sometimes only just a little bit better in terms of mAP in no-context setup, which could mean that the model is more confident within no-context approaches); context really makes a difference in the gated pooling approach, where it noticeably improves the performance of both models. Whisper model seems to be the model which hardly relies on context within the PII multilabel classification task with frame input size; results for the Whisper model remain poor even with context, but without it, it seems meaningless at all(see 6.2 for exact metrics output).

5.1.5 Summary Multilabel Results

After reviewing results for every input size and different pooling techniques, we can confidently say that different input size shows different results and require different combinations of pooling techniques and models for reaching the best performance for this particular demo data.

For the phrase approach, the best combination appeared to be Whisper and the multihead attention technique; however, we also noticed that with other pooling techniques, Whisper does not perform that well. Whisper still demonstrates comparable performance to the WavLM approach, which was consistently good and performs relatively the same for all pooling techniques. Hubert appeared to be the worst approach for the phrase level; in all three pooling approaches, it showed consistently poor performance, and the results in general were almost the same with different pooling techniques.

For the Word level input top scorer remained the same - Whisper with multihead attention pooling. But unlike in the phrase approach, where Whisper showed good performance with every pooling technique in the word approach, multihead attention appeared to be the most stable and decent performance, with F1 micro = 0.52 and F1 macro = 0.48; meanwhile, other pooling techniques did not have even close to these results with Whisper. WavLM, as well as the phrase input window, again showed decent performance among all pooling techniques and got even closer to the current leader with F1 macro = 0.35 and F1 micro = 0.93 in a multihead attention setup. With other pooling techniques for word-level, WavLM

seemed to be a strong leader with solid performance, which is close to multihead attention results.

An equal slice input window, Whisper finally gave up its leadership to other setups. Whisper seemed to struggle to classify PII in an equal slicing setup; it did not manage to show comparable performance with previous input window setups, and did very poorly with every pooling technique(the results were almost identical). Meanwhile, WavLM + hierarchical pooling took first place as the best PII classification model for this particular input size, with a solid performance and F1 macro = 0.39 and F micro = 0.94. Regarding other pooling techniques in combination with the WavLM model, they are almost as effective as the hierarchical one and keep maintaining the trend that the WavLM model performs on average the same with different pooling techniques represented in this research.(Multihead attention is second best for WavLM, and Gated is the worst, with a noticeable difference in terms of F1 macro, which is equal to 0.3166). Hubert, with an equal slicing input window as well as the WavLM model, demonstrates decent performance; it is consistently worse than WavLM but also always better with any pooling technique than Whisper. The pooling technique for the Hubert model also does not really affect the results of the performance.

Finally, frame-wise input results were also reviewed in this section. Here, the best performance was demonstrated by a combination of the WavLM model and the Multihead attention pooling technique. The frame-wise case was the only one where the WavLM model performance depends on the pooling technique and demonstrates a noticeable difference between Multihead attention pooling, Hierarchical and Gated (Anyway, the difference between gated and hierarchical pooling was not that crucial). The result of the WavLM model was not only the best in the frame-wise input window case, but in general for WavLM among all input sizes and pooling techniques. Whisper, meanwhile, remains the worst model in frame-wise as well as it was in the equal slice approach, and none of the pooling techniques could prevent the degradation of performance, and all of the results within the 3 pooling techniques are relatively the same. Hubert, as well as WavLM, only improved its performance with smaller input sizes, and the frame-wise case is not an exception. With the Hubert frame-wise case, we can also observe that pooling matters and the best result was managed to be achieved with multihead attention pooling; hierarchical pooling in combination with Hubert demonstrated the best performance within the frame-hierarchical domain; and the worst pooling for Hubert was the gated one.

In general, the results of conducted experiments confirm that different input sizes, different input windows and different pooling techniques

in combination with each other can affect the general performance in a multilabel classification setup. With a certain amount of confidence, we can say that the Whisper model is better for a wider input window and performs the best on the phrase and word level with exact time stamps. However, WavLM was the most stable model in terms of performance and consistency, regardless of pooling technique and input size; the only significant difference in performance that was mentioned based on different pooling techniques was in the frame input window domain. Overall, WavLM was not only the most consistent but also the best for equal slice and frame-wise approaches. Hubert, as expected, was never the best model among the 3 chosen models; however, it was noticed that the Hubert model improves its performance as the input size gets smaller. All further hypotheses and possible reasons why models perform that way will be provided in the discussion Chapter 5.7

5.2 Binary results

In this section, the results for the binary PII classification approach will be provided for the same four input sizes and compared to the performance of the Multilabel domain in terms of settled metrics. Results will be analysed mainly based on F1 score; further discussion regarding possible reasons why models and pooling combinations perform this way will be provided in the discussion Chapter 5.7, as well as for the Multilabel approach.

5.2.1 Binary Phrase input window

Table 5.5 shows results for binary classification with Phrase window input, and already some differences could be mentioned in comparison to Multilabel Phrase input results, which were described in Table 5.1.

Table 5.5: Binary phrase-level PII classification results for different models and pooling strategies.

Model	Pooling Strategy	Precision	Recall	F1
WavLM		0.6207	0.4417	0.5242
HuBERT	Multihead-Attention	0.3828	0.5542	0.4528
Whisper		0.5117	0.6258	0.5630
WavLM		0.5767	0.5072	0.5397
HuBERT	Hierarchical Pooling	0.4125	0.4049	0.4087
Whisper		0.2933	0.3845	0.3327
WavLM		0.3909	0.5887	0.4698
HuBERT	Gated Pooling	0.3734	0.4943	0.4254
Whisper		0.2882	0.5399	0.3758

In the table above, we can see that, as well as in Phrase input in the multilabel domain Whisper model again becomes the best in terms of F1 score. From the results, we can also see that for Whisper with Phrase window input in the binary classification task, the pooling technique is important, and the best result was demonstrated with multihead attention pooling, and the worst one was in combination with hierarchical, gated was slightly better than hierarchical, but still significantly worse than the multihead attention one.

WavLM also demonstrates stable performance among all pooling techniques and almost has the same result as the Whisper model with multihead attention pooling. The F1 score is almost identical for both hierarchical and multihead attention pooling and approximately equal to 0.53; the only difference in F1 score for Wavlm is observed with gated pooling, where results are slightly worse with F1 score = 0.4698. In general, WavLM seemed to perform the same way with different pooling techniques in the binary setup as in the multilabel one.

Hubert is the model which demonstrates a significant difference in terms of results in comparison to the Phrase input in a multilabel setup and demonstrates performance comparable to the results of WavLM and Whisper models. The best pooling technique for Phrase window input appeared to be the multihead attention one. However, it seems that different pooling techniques affect the Hubert model less than other models. Within other pooling techniques, Hubert shows more or less the same performance and even outperforms Whisper in Hierarchical and Gated pooling approaches.

Context also has an influence on phrase input size. WavLM within the no-context setting outperforms context setup in every pooling technique, with F1 scores equal to 0.5889, 0.5701, 0.5683 for multihead attention, hierarchical and gated poolings, respectively. Hubert, meanwhile, showed more or less the same results, and the context does not seem to improve or degrade the results of the model. For the Whisper model, context appeared to be very important with multihead attention pooling and improved f1 score from 0.48 without context to 0.56 with it; performance also improved with context within hierarchical pooling, but within the gated pooling setup model seems to be slightly better without context. (However, both hierarchical and gated setups perform very poorly in general, especially in comparison to the multihead attention setup and to other models' performances)

The results of Phrase window input in the binary approach clearly demonstrate that models are able to successfully classify PII based on audio input. Overall results are relatively similar to the Multilabel results for the same input size; however, we can definitely see that for the Hubert model, it is way easier to classify within the binary domain than within the multilabel one.

5.2.2 Binary Word input window

Table 5.6 shows results for the Word input window within the binary classification task, and at first glance, Whisper seems to be the best model in terms of performance again.

Table 5.6: Binary word-level PII classification results for different models and pooling strategies.

Model	Pooling Strategy	Precision	Recall	F1
WavLM		0.4225	0.4380	0.4301
HuBERT	Multihead-Attention	0.1834	0.8554	0.3020
Whisper		0.6180	0.6394	0.6285
WavLM		0.4383	0.549	0.4874
HuBERT	Hierarchical Pooling	0.1809	0.8496	0.2983
Whisper		0.0564	0.3936	0.0987
WavLM		0.3909	0.5887	0.4698
HuBERT	Gated Pooling	0.1842	0.8480	0.3026
Whisper		0.0594	0.3632	0.1021

As well as in the Phrase input window, both for multilabel and binary approaches, the best combination of model and pooling technique appeared to be Whisper with multihead attention pooling. However, it is clear that with this particular input size, Whisper appeared to be really dependent on the pooling technique since the results between Multihead attention and other pooling techniques differ dramatically. Within the multilabel classification task for word window input, we have already seen this kind of degradation in performance, so this is not a surprise.

WavLM again demonstrates stable performance among all Pooling techniques; however, now the best pooling technique in combination with the WavLM model appears to be Hierarchical Pooling, as well as in a multilabel approach, with an F1 score = 0.4874. Gated pooling in this case even managed to outperform multi-head attention pooling (f1 = 0.4301) with an f1 score of 0.4698; meanwhile, in a multilabel classification task, multi-head attention pooling performs better than gated one.

The Hubert model has performed consistently decent in combination with every pooling technique in a binary classification setup. It is still outperforming the Whisper model within Hierarchical and gated pooling, but the results remain not so great, with F1 scores approximately equal to 0.3. In comparison with multilabel results, binary results are more stable across pooling techniques than multilabel one, where Hubert's performance differ depending on the pooling approach.

Context influence is also noticed within word input size results. WavLM performs consistently better within multihead attention and hierarchical

pooling, but within the gated pooling context improves performance. Hubert’s results seem to be the same both with and without context inputs. Whisper model within word input size setup has better results in the no-context setup with F1 score equal to 0.73; however, in other pooling techniques, results do not differ, and the F1 score is consistently poor both in no-context and context setups.

In the binary classification domain with a word input window, we can see that Hubert improved in terms of F1 score in comparison to the multilabel domain and performs far more stably. Meanwhile, Whisper struggles more within the binary domain, and performance degrades much more than in the multilabel setup if gated or hierarchical pooling is used. WavLM remains the most stable and consistent model, which manages to demonstrate decent performance with a word input window regardless of the pooling technique.

5.2.3 Binary Equal input window

Table 5.7 demonstrates results for an equal slicing input window within the binary PII classification task, and from the table, we can already see that it significantly differs from the results for multilabel classification, which were presented in table 5.3.

Table 5.7: Binary equal-slicing (0.5 s) PII classification results for different models and pooling strategies.

Model	Pooling Strategy	Precision	Recall	F1
WavLM		0.2685	0.3951	0.3197
HuBERT	Multihead-Attention	0.1770	0.1733	0.1752
Whisper		0.0302	0.2154	0.0529
WavLM		0.2402	0.4005	0.3003
HuBERT	Hierarchical Pooling	0.1310	0.2970	0.1818
Whisper		0.0312	0.2589	0.0557
WavLM		0.2123	0.4743	0.2933
HuBERT	Gated Pooling	0.1566	0.3067	0.2073
Whisper		0.0279	0.5968	0.0534

The best combination of pooling technique and model for equal slice input window appeared to be the WavLM with multihead attention pooling.

The WavLM model in a binary setup with equal input, 0.5 sec remains stable in terms of metric and maintains an F1 score close to 0.3 among all pooling techniques within the binary classification task.

Whisper performed very poorly among all pooling techniques in combination with the model, and the F1 score for multihead attention, gated and hierarchical poolings all equal to approximately 0.05. Meanwhile, for multilabel classification Whisper model demonstrated more stable results in terms of F1 macro, which was equal to approximately 0.2; it is still poor, but in a face-to-face comparison, Whisper, on average, performed better in a multilabel set-up.

The influence of context is not really noticeable in the WavLM setup among all pooling techniques in the slice classification setup. For the Hubert model, we can see from Appendix 6.2 that results with context are just slightly better than without it. Whisper completely failed in the equal slicing setup, and the results appeared to be very poor both within context and no-context setups.

Hubert, as well as WavLM, performs more or less stable within the binary domain, and the F1 score ranges from 0.17 to 0.20 depending on the pooling technique. Despite the performance of the Hubert model still being poor in terms of metrics among all pooling techniques, it still outperforms Whisper in hierarchical and gated pooling.

5.2.4 Binary Frame input window

And finally last set of training processes was conducted with frame window input within the Binary PII classification task. In Table 5.8, the results for frame-wise binary PII audio classification can be found.

Table 5.8: Binary frame-wise PII classification results for different models and pooling strategies.

Model	Pooling Strategy	Precision	Recall	F1
WavLM		0.1835	0.8876	0.3042
	Multihead-Attention	0.4637	0.6000	0.5231
	Whisper	0.0515	0.2699	0.0865
HuBERT		0.1929	0.8859	0.3168
	Hierarchical Pooling	0.3160	0.6120	0.4170
	Whisper	0.0562	0.2777	0.0935
Whisper		0.2508	0.3719	0.2996
	Gated Pooling	0.3830	0.5510	0.4520
	Whisper	0.0559	0.3659	0.0970

Surprisingly, the best performance on the frame input level is demonstrated by less advanced Hubert model. Moreover, it is not only better within multihead attention but outperforms Whisper and WavLM models in any other pooling technique. Meanwhile, in a multilabel setup, Hubert was trying to compete with WavLM but was still far away. In a binary setup, it demonstrates confident performance with an F1 score equal to 0.52 in multihead attention and outperforms WavLM by 0.2 points.

WavLM in the binary setup takes second place and remains as stable as before for all input sizes, pooling techniques and different setups. The F1 score for the frame-wise PII classification varies from 0.29 to 0.31. However, binary results for frame input size slightly differ from the multilabel one, where it performs noticeably better with multihead attention pooling; meanwhile, in the binary setup, it performs relatively the same among all pooling techniques.

Whisper completely fails in the task of binary frame PII audio classification, and the results in terms of F1 score are very poor and vary from 0.08 to 0.09. In general, these results replicate the results with a multilabel setup and prove that Whisper performs worse with a narrower input window.

Context influence is also noticed within the frame domain, but not for all setups. Within WavLM setups with different pooling techniques, only in the hierarchical pooling technique difference in performance between context and no-context results is observed, but only in a range of 0.03 in favour of the context approach. Meanwhile results of Hubert Performance

are noticeably better with context for the multihead attention pooling approach(F1 no-context = 0.459), but with the Hubert hierarchical pooling combination model performed better without additional context, with an F1 score equal to 0.47 in that case; Hubert, in combination with gated pooling, performed way better with context. Whisper consistently performs badly both within no-context and context setups and performs almost identically poorly within those settings.

To summarise, the results in Table 5.8 show that, as in the multilabel classification setup, the frame input remains the most challenging input size for the Whisper model. However, the results for the frame input window also confirm that HuBERT performs noticeably better with smaller input sizes and even outperforms more architecturally advanced models across all pooling techniques.

WavLM, despite its generally strong and consistent performance across pooling methods, performs significantly worse than HuBERT. This is somewhat surprising given the architectural differences between the two models. One additional observation can be made when comparing the multilabel and binary setups: the combination of WavLM with multihead attention improves performance in the multilabel scenario, whereas in the binary case, pooling has no meaningful effect on the model’s performance.

5.3 Summary for Binary results

Reviewing the results for the PII binary classification task has provided us with some valuable insights and has also allowed us to notice that some of the tendencies in the model’s performance results are applicable both for multilabel and binary classification tasks.

For the Phrase input window, the best combination of model and pooling technique appeared to be Whisper with multihead attention pooling, as well as in the multilabel classification task within the same input size. However, in the binary classification setup, pooling technique appeared to be much more important since the Whisper model performs good only with multihead attention. In the multilabel setup, performance also degrades for the other two pooling techniques, but not that dramatically, which could be a sign that the pooling technique could be a crucial parameter in a binary setup, in particular. WavLM and Hubert models, in the meantime, have demonstrated decent and consistent performance within the binary PII classification task for phrase input size.

Word input size results for the binary setup also look familiar in terms

of performance to multilabel results. The best performance for word input size was demonstrated by the combination of Whisper and the Multihead attention pooling technique. As well as in a multilabel setup, the performance in binary only decent with multihead attention pooling, meanwhile other gated and hierarchical poolings completely fail and demonstrate very poor results within the binary classification task for word input. The first difference between binary and multilabel tendencies in the results appeared within the Hubert performance with word input size. Despite being within a binary PII classification task, Hubert has stable results with every pooling technique, with the value approximately equal to 0.30 among all model+pooling combinations. This shows that different pooling techniques don't give a significant performance advantage within the binary classification task. Meanwhile, in the multilabel results for the word input window, we can see that the pooling technique matters and the combination of Hubert and multihead attention pooling shows much better results than hierarchical and gated pooling in the same task. WavLM within the binary classification task demonstrated the most consistent and stable performance among all pooling techniques and models, with an F1 score within the 0.43-0.48 range. The best result for WavLM within the binary classification task was shown in combination with the Hierarchical pooling technique; the same was in the multilabel case, if we compare to F1 macro and taking into account the mAP metric, which reflects the confidence of predictions; however, multihead attention appeared to be the worst pooling technique within the word binary PII classification task.

For equal input size within binary classification, the best model appeared to be again Whisper in combination with multihead attention pooling. Other pooling techniques demonstrated very poor results with an F1 score equal to 0.05. In comparison to multilabel results, within the binary domain Whisper model appeared to be less stable and relied more on the pooling technique; meanwhile, in the multilabel setup, it still demonstrates poor but more consistent results. WavLM continues its trend and, as usual, remains the most stable model and demonstrates more or less the same result(approximately 0.30) within every pooling technique with the binary PII classification task. In a multilabel setup, WavLM meanwhile has a broader distribution in terms of metrics and depends on pooling more in the binary setup. Hubert model, as well as WavLM, demonstrates similar results within all pooling techniques, ranging from an F1 score of 0.17 with multihead attention to 0.20 with gated pooling. In a multilabel setup, Hubert performs better with a multihead attention setup, while the gated pooling technique appeared to be more suitable for a binary classification task.

Results for the frame input window within the binary PII classification

task appeared to be the most exciting and interesting. Hubert, in combination with multi-head attention pooling, demonstrated the best results; other pooling techniques also worked well for frame-wise classification, with F1 scores equal to 0.41 and 0.45 for hierarchical and gated poolings. (Within the multilabel setup, the Hubert model had demonstrated decent performance; however, the results in the multilabel setup were average and not even close to being the best. Meanwhile, within a binary setup, it absolutely outperforms other models by far.) WavLM was observed as worse than the Hubert model, but much more stable and consistent and confirms again that the pooling technique is not a crucial part of the model’s performance in the task of binary PII classification. However, in the multilabel classification setup, pooling techniques had a stronger impact on performance. In particular, combining multihead attention pooling with WavLM yielded the best results for frame-wise multilabel PII detection. In contrast, in the binary setup, pooling methods produced nearly identical performance.

5.4 Result Summary

The section above and analysis of Multilabel and Binary approaches for PII classification task with different pooling techniques showed that the results depend on the input size, pooling techniques and models.

In the Table 5.9, the best combinations of model and pooling techniques for each input size are provided. In the table, both results for binary and multilabel setups are mentioned.

Table 5.9: Summary of best-performing approaches for each input size in binary and multilabel setups.

Input Size	Best Binary Approach	Best Multilabel Approach
Phrase-level	[WavLM + MHA NC F1 = 0.58]	[Whisper + MHA mAP = 0.44 / F1(micro) = 0.76]
Word-level	[Whisper + MHA NC F1 = 0.73]	[Whisper + MHA NC mAP = 0.69 / F1(micro) = 0.67]
Equal-slice	[WavLM + MHA F1 = 0.31]	[WavLM + HR NC mAP = 0.33 / F1(micro) = 0.94]
Frame-level	[Hubert + MHA F1 = 0.52]	[WavLM + MHA mAP = 0.56 / F1(micro) = 0.94]

Notes: MHA = Multihead Attention Pooling; NC = No Context. HR = Hierarchical Pooling

From the table above, we can see that all of the models find their place somewhere within input sizes to shine. In most sizes, WavLM with multi-head attention pooling appeared to be the best model. Whisper performed very confidently within Phrase input size in Multilabel setup; within Word setup, Whisper dominated in both Multilabel and Binary settings. Within a multilabel and binary setup with equal slices, input size, WavLM

appeared to be the best model among others, and the curious thing is that this is the only time when Hierarchical pooling outperforms Multihead Attention one (but only in the multilabel setup). WavLM also became the best model for frame-wise multilabel PII classification. Hubert, in combination with Multihead attention pooling, became the best model in the Binary approach; meanwhile, in a multilabel frame-wise setup, the best model appeared to be WavLM, also with multihead attention pooling.

The results reflect a clear trend that Multihead Attention (MHA) is the most relevant pooling technique among others, both for binary and multilabel audio PII classification tasks. Also, results confirming earlier mentioned hypotheses that Whisper will be more suitable for wider input windows but will completely fail within frame-wise input size, which was confirmed by results in Tables 5.8 5.7 and 5.4, where Whisper lost to all other models + pooling combinations.

From the results in the Table 5.9, we can also see that the context influence is obvious in some of the setups, and sometimes context improves performance, sometimes not. For the solutions where the input is predefined by transcription timestamps (Phrase and Words), in most cases, context appeared to be an obstacle and confused the model, making no-context solutions better (Only the multilabel phrase setup context appeared to be useful to significantly improve the results). When it comes to approaches with other logic where we have not used predefined time stamps from transcriptions (equal slicing and frame level), context played some role and helped to improve the performance of the models.

To sum up, the section with results helped not only see how good different models perform with different input sizes and pooling techniques, but also to figure out the best setup which will be trained with the full amount of data. This section also showed that the results in some of the approaches heavily rely on the context of input data, which will be further analysed in Chapter 5.7.

Due to computational cost, it will be challenging to train every setup on the full amount of annotated data, so the decision was made in favour of the best multilabel frame-wise approach. The training on the full amount of data can both improve the performance of the model or make it worse; however, the results will highlight the model's ability to generalise. More detailed results and comparison of models, which will be trained on the full amount of data to Golden Standard, Presidio Annotation and Hubert Binary setups, will be provided in the following sections.

5.5 Full dataset training Results

According to previous sections WavLM Multihead attention pooling approach for multilabel classification appeared to be the best among other solutions; therefore, this model was chosen to be the one which will be trained on the full amount of data that we have collected and annotated previously.

Table 5.10: Performance summary of the model.

Metric	Score
mAP (macro)	0.4049
AUC (macro)	0.8721
AUC (micro)	0.9287
F1 (macro)	0.4183
F1 (micro)	0.8059

Table 5.10 shows results for the model, which were Best in the multilabel setup with frame input. From the results, we can see that the model performance with the full amount of data did not turned out to be advantageous in terms of metrics, and we can see noticeable performance degradation. However, despite results not being as good as they used to be in the demo approach, the performance remained decent, and it even outperformed almost all other model+pooling combinations within frame input size settings (except HUBERT with multihead attention, which is still slightly better).

The model was fine-tuned with the same hyperparameters on 10 epochs with BCE loss and the same AdamW optimiser. The training and evaluation process together took a bit more than 120 hours.

5.6 Base-line vs. Presidio vs. Custom approaches

In order to evaluate the model’s performance more accurately Gold Standard transcription of the audio file was achieved by manual annotation (more details about the gold standard can be found in the dedicated section 4.3).

In this section, we will compare the Presidio model performance on the gold standard against that achieved in the previous section, only for audio setups. The comparison of the Presidio annotation to the Gold Standard will help to determine how accurate was corpora used in this research. Also, we will use audio together with the silver standard version

of the gold standard, which was achieved by automatic annotation with Presidio, to see how the model performs with unseen data. After that, we will evaluate performance with gold standard annotation and see if the results are improving or degrading. These tests will not only provide us with results on how the model performs with unseen data but will help to find out whether manual annotation is crucial and how silver corpora are incorrect in terms of accuracy of annotation, and verify that the proposed approach in this research work is sane.

Table 5.11: Comparison of Multilabel and Binary models on unseen data.

WavLM + MHA (Multilabel)	HuBERT + MHA (Binary)
Multilabel Results:	Binary Results:
mAP (macro): 0.4013	Precision: 0.3651
AUC (macro): 0.8536	Recall: 0.6013
AUC (micro): 0.9007	F1 Score: 0.4543
F1 (macro): 0.3675	
F1 (micro): 0.7564	

Table 5.11 shows the results of two models for the silver version of the gold standard data. For the multilabel approach WavLM model with Multihead attention was used, fine-tuned on all data from our corpus. For the Binary approach, we used the best model for the binary approach with frame-wise input - Hubert + multihead attention.

Binary model results show that the model performs pretty well on unseen data in comparison to the test results that we have seen before in Table 5.8, where the F1 score result for this approach was equal to 0.52, whereas on unseen data, it is equal to 0.45. Despite performing a bit worse, it is still a meaningful result.

For multilabel settings where the model was trained on full data (Table 5.4), it does not perform that well on test audio. Performance in a multilabel setting is less confident, and we can see it reflected in all of the metrics. Despite the results being worse, it is still a meaningful performance.

However, we should keep in mind that there are still results for silver-annotated data. Further, you will see the results of model performance for the same audio file, but with Gold Standard annotation.

Table 5.12: Comparison of Presidio, Binary, and Multilabel models against the Gold Standard.

Presidio	Hubert MHA	WavLM MHA
Presidio vs Gold:	Binary vs Gold:	Multilabel vs Gold:
Precision: 0.9605	Precision: 0.4841	mAP: 0.2592
Recall: 0.6986	Recall: 0.5837	AUC (macro): 0.6581
F1 Score 0.8089	F1 Score: 0.5293	AUC (micro): 0.7287
		F1 (macro): 0.3056
		F1 (micro): 0.5597

Table 5.12 above shows the results of the classification of the same audio file in comparison to the Gold Standard transcription. It is worth mentioning that the Presidio model was used on the Gold Standard transcription text directly; meanwhile, other models were tested on audio, and then the output was compared to the Gold Standard.(Since Presidio is obviously a text model)

From the Table 5.12 we can see that Presidio, as expected, was not perfect with silver annotation. The results of the F1 score equal to 0.8 prove that automatic data annotation and PII tagging of audio could have imperfections, and we need to keep in mind this disadvantage of our approach. However, this is still a very decent performance (most of the misclassifications happened when Presidio considered schools and universities as non-identifiers).

For binary Hubert+Multihead attention pooling, we can see an improvement in performance. Hubert's predictions have more common labels in comparison to the Gold transcript than in comparison to the Silver one for the same audio. The improvement of performance in that case can mean only that the model can predict some classes which Presidio was not able to do during the PII Tagging stage (In most cases Hubert approach managed to predict school names as PII, meanwhile Presidio considered them as non-identifiers, this could be a reason of performance improvement for the binary audio approach which has worse results with silver annotation but better with gold one).

In the Multilabel WavLM + Multihead Attention Pooling configuration trained on all data from our corpus, we observe that the results on the Gold Standard transcription are slightly worse than those obtained with Silver Annotation for the same audio files. Although the model appears less confident according to some metrics, this may be attributed to issues in the Gold Standard annotations and the absence of detailed annotation

guidelines, as discussed in Section 4.3.

We also noted that the most frequently confused classes during both training and testing were **LOCATION** and **PERSON** (excluding the **NOIDENTIFIER** class, which is naturally more prone to confusion due to its substantially larger number of instances). This pattern mirrors the tagging errors made by Presidio, as illustrated in Listing 4.2. Suddenly, our audio-only classification model likely replicates some of the same mistakes observed in the text-based Presidio approach. However, we cannot firmly conclude this without additional data for validation. This could be one of the directions in future research.

Furthermore, the **NRP** class proved the most difficult for the model to predict, frequently being misclassified as **NOIDENTIFIER**. This issue may stem from both the limited number of **NRP** examples in the training set and from the complexity of this category, which requires distinguishing between political affiliation, religion, and nationality in the same class. Manual inspection of the classification outputs revealed that nationality was generally predicted accurately and consistently, whereas political and religious attributes remained challenging. But those observation also requires additional validation with more data and deeper research towards this direction. Increasing the number of training instances and potentially splitting **NRP** into three separate categories - nationality, religion, and political views - may improve performance.

Aside from **NOIDENTIFIER**, the class most reliably predicted by the model was **DATE_TIME**.

5.7 Discussions

5.7.1 Pooling techniques influence

From the results that were mentioned in the above Sections of Chapter 5, we can see that across different input sizes, models and setup settings we can see the pooling technique plays a noticeable role in model performance.

In general, Multihead Attention pooling appeared to be, on average, the most universal Pooling technique among all input sizes and Model setups. In most cases, exactly multihead attention pooling improved the performance of the models within all input sizes and both in multilabel and binary setups. So this technique could be easily considered as the most

universal and suitable for the task of PII classification with only audio features. In multiple different works, Multihead Attention pooling also appeared to be the key factor in audio classification tasks, for example, in (Wang et al., 2019), authors mention Multihead attention pooling as the best pooling technique for audio scene classification tasks. Another work that highlights the Multihead Attention pooling advantages over other pooling techniques is (Hong et al., 2020) where Authors confirm that gated pooling performer worse then attention pooling within classification task; Authors also used combination of gated pooling and Multihead Attention pooling and discovered that such combination significantly boost performance in terms of metrics and increasing mAP and F1 score within audio classification task.

Hierarchical pooling was not really consistent in terms of performance and completely failed in combination with the Whisper model in both multilabel and binary setups. However, it still performs decently with Hubert and WavLM models(this is explainable because both share common architecture, as was said before in Section 4.7). However, despite being average in terms of performance, it has managed to be the best pooling technique for the multilabel classification task within an equal slice input size. However, as said before, this is the most controversial size, which, with a variety of trade-offs, is not really worth it with a frame-wise approach.

The last pooling technique which was used in this research is the Gated pooling technique. This one appeared to be the worst pooling technique and consistently failed among all input sizes, models and approaches combinations. In the already mentioned paper (Hong et al., 2020), authors mention that the gated mechanism could be really rigid and suppress useful information; in the setting of extremely unbalanced data, this became a huge bottleneck for this pooling technique, and this is probably why the model performed so poorly in combination with gated pooling.

5.7.2 Input size trade-offs for models

All of the models in this research have shown the ability to work with different input sizes to some extent, but it is clear that all of them have ability to classify identifiers in our dataset. Earlier, in the Chapter 5, we also mentioned that certain models work better with certain input sizes.

The Whisper model appeared to be the best model for wider input size, but only with multihead attention pooling. In narrower sizes, Whisper struggles to have reasonable performance despite of setup,

pooling technique or context. Also, Whisper was extremely good with binary setup and word input size, which also confirms that it easily captures the semantic nature of audio input on a wider input level. In one of the research papers (Gong et al., 2023), the authors mention Whisper’s generalisation ability and highlight its robustness in the task of audio tagging. Another work confirming Whisper’s ability to classify audio inputs is (Ameer et al., 2025), where authors used Whisper as an encoder for multilabel classification of speech disfluencies. All of these studies confirm that Whisper can capture information both semantically and tonally from the input data, which makes it a strong choice for PII classification tasks, but according to the results obtained in this research, this will work only within wider input sizes like phrases or whole words.

WavLM, unlike Whisper, works more stably with any input size proposed in this research. Also, the model’s architecture allows to have decent results in all within polling techniques. WavLM also shows that it works well for both narrow and wide input windows, as well as both in binary and multilabel setups. As mentioned in (Chen et al., 2022), WavLM is good at both segment and frame speech processing tasks and achieves the top score in the speaker verification task. Also, in this paper authors mention that models at the moment have demonstrated state-of-the-art performance among multiple speech processing tasks at different spanning levels (speaker recognition, speech command recognition, emotion classification). This paper confirms the universal and versatile nature of the WavLM model for speech processing tasks; the result that we have obtained with our setups supports this statement.

The Hubert model performs decently among all input sizes which were used in this work. However, we noticed the trend in the results section that the model performs better with smaller input sizes, both in multilabel and binary setups. This trend also finds its recognition in other works within the field of audio classification. For example, in (T.-Y. Wu et al., 2022), authors highlight Hubert’s abilities to capture frame-level acoustic features and also mention that the performance degrades with longer temporal context(wider inputs). Results of Hubert performance with different input sizes in our research also correspond to the original Hubert paper (Hsu et al., 2021), where authors mention that the model’s design favours frame-wise input or short segment classification tasks. The model’s architecture peculiarity is reflected in the results of our research, where it has performed better than every other setup on the frame-wise input level within the binary setup.

5.7.3 Context influence on Model's performance

Another question that we have attempted to answer in this research work is whether context with input will help the model to make predictions, and if it can build relations between the context and the target input.

Whisper was the only model which demonstrated consistent improvement with contextualised input in terms of F1 scores and mAP. Even in wider inputs like phrases, where it already has more data to operate, 0.5 seconds of additional context to phrases improved the performance further. Original authors of the Whisper approach, mentioning in their paper (Radford et al., 2022) that Whisper has a receptive field of 30 seconds, and this is way above the phrase input and even dramatically more than frame one. This explains why it has such a good performance with bigger input sizes but completely fails with frame one. Our results show that the Whisper model could be used for the audio PII classification task, and it performs very confidently, but only with bigger input sizes.

WavLM performs stable across all input sizes with and without context, but really noticeable improvements metric-wise could be noticed within frame and equal slice input sizes. In the previously mentioned paper (Chen et al., 2022) authors mentioned that the architectural design of the model allows to be more precise with additional context(in this particular paper model demonstrates better results in tasks of speaker verification, speech separation, and diarisation). So we definitely can say that the results and their context-wise trends correspond to already existing papers' results, which aligns with our findings for WavLM.

Hubert was the model in this research results of which are less likely to be affected by context. This tendency could be explained by referring again to the original Hubert paper (Hsu et al., 2021) where authors explicitly mention the model's designed to be used on frame-wise tasks. Another work which can support the results of our work is (Meng et al., 2025), where the authors mention that the Hubert model operates well in streaming mode and make it clear that context beyond 200ms to 500ms makes the performance of the model worse. These papers confirm our results and statement that Hubert works better with local acoustic input than with longer phrase-level dependencies.

In general, the results of our research confirm that integration of context into the model's inputs can be useful. However, the results also show that depending on the input size and the model's architecture, we should adjust the context length accordingly.

Chapter 6

Conclusion

Having tested different models and their configurations for the PII classification task within the framework of our approach and the prepared dataset, we can finally provide answers to the research questions posed in this study.

1. **RQ1:** How accurately can models identify the presence of PII in speech based only on audio features input?

Answer: In Chapter 5, we evaluated the proposed approach and discussed its outcomes. Based on the results, we can confidently conclude that PII classification from audio alone is feasible in both binary and multilabel settings, at least for our dataset. The models achieve meaningful performance and clearly demonstrate the ability to distinguish PII from non-PII instances.

2. **RQ2:** How do different encoder-based models (Whisper, WavLM, and HuBERT) perform with different audio input sizes?

Answer: In Section 5.7.2, we observed that different models perform optimally with different input window sizes. Each model's architecture naturally favours certain input dimensions. Whisper is the clear leader for larger input windows, while HuBERT performs better with narrower ones. WavLM appears to show the most stable performance overall: although it falls short of Whisper on setups involving wider inputs, it delivers the most consistent performance across all input sizes, not just the larger (phrase, word) or smaller ones (equal-slice, frame-wise).

3. **RQ3:** How can additional context to the inputs affect model performance?

Answer: In Section 5.7.3, we analysed the importance of context and found that, in some cases, it can indeed have a positive impact on performance, while in others it may actually confuse the model. Overall, we cannot make definitive claims about the effect of context because we evaluated only a single context size of 0.5 seconds. What is clear, however, is that context does influence performance, though the nature of that influence remains an open question and a promising direction for future research.

4. **RQ4:** To what extent can audio-based PII detection operate as a multilabel classification problem (distinguish between different types of PII)?

Answer: At the end of Section 5.4, we discuss the model's behaviour in a multilabel classification setup. For this corpus in particular, the models were able to reliably distinguish PII from NoPII and demonstrate the ability to distinguish one class from another as well. They showed some difficulty with the less represented class(NRP), which may stem from the design of the PII annotations and the broad definition of identifiers within the NRP category. In contrast, the models were generally confident when identifying the DATE_TIME class. Overall, the encoder-based models demonstrated the ability to differentiate between classes in multilabel settings of the PII classification task with only the audio inputs, and the results presented in Chapter 5 support this conclusion.

6.1 Limitations

In this section, we will outline the limitations of this work.

Firstly, it is important to underline the nature of the data. Since the corpus was automatically transcribed with Whisper, the construction process may have introduced errors. Although we have used a state-of-the-art Whisper V2 large model, there could still appear some imperfections both in word transcription and in the temporal alignment of the words and phrases.

Another bottleneck of this work is the tagging of the identifiers, which was performed with Presidio. As we saw during the comparison of the Presidio annotation to our Gold one (5.6), the automatically parsed corpus has some imperfections. Interestingly, our custom approaches for PII classification appeared to perform better in comparison to the Gold transcription within a binary setup.

Presidio itself could also be one of the limitations of this work. The

choice of the model for automatic PII tagging in the preprocessing part of our approach can affect the results significantly, which was mentioned in Section 5.6, where Presidio’s inaccuracy during the preprocessing part was revealed with the help of the Gold Standard. According to recent research within the field of PII detection in text (Savkin et al., 2025), with Presidio, precision is often lower than recall, indicating many false positives. Thus, Presidio NER+rule-based architecture can sometimes be rigid and not able to capture more difficult types of identifiers or confuse them (and this was confirmed during manual annotation of the Gold Standard). More modern methods, LLM-based ones, could be a better choice and improve the results of the annotation part.

The loss function is another limitation of this work. BCE loss function, which we used during the training process of models, despite being a universal solution both for Binary and Multilabel setups, has its own disadvantages within the frame of this research paper. In (T. Wu et al., 2021) authors mention that BCE loss decomposes multilabel problems into independent binary tasks; therefore, it can not capture inter-label dependencies, especially in cases with unbalanced data. Anyway, BCE loss remains a decent choice for the Binary approach.

Another limitation that we face in this work is the domain composition. Although we have empirically proved in Section 4.4.4 that our data is sane and contains relatively the same distribution between PII and NoPIIs in comparison to other corpora within the PII classification task, we still lack diversity in terms of the interview domain, since all of our interviews lie within the domain of Narrative Interview.

The small size of Gold Standard transcription seems to be another limitation of our work. Since we have only one manually annotated transcription, the results, despite being insightful, are still not really reliable and representative. Providing more instances of precisely annotated data can solidify results and improve the evaluation part of our approach.

One more limitation of this work is the size of the models, parameter-wise. In our research we use relatively small model sizes with approximately 75-90 million parameters for speech PII classification. With bigger model sizes, results probably could be improved.

Context size is another factor that should be considered in future work. In our experiments, we tested only a single context setting of ± 0.5 seconds for each input, which does not provide enough variation to draw confident conclusions about its impact. Therefore, exploring a wider range of context sizes should be one of the possible further directions.

In addition, we have only conducted all experiments on one seed (42), which is another limitation. Conducting experiments on a different seed could provide different results.

6.2 Future directions

We believe that the results obtained in this research could be useful in future work within the field of speech anonymisation and PII classification, with only audio features in particular.

The results of the approach proposed in this paper showed that it can be used in other works and with other datasets and models, despite the inaccuracies associated with automatic transcription and PII tagging. Results in Chapter 5 show that models are able to detect PII based only on audio features of input.

We also noticed that several factors matter in the task of automatic PII detection. First of all, input size and model selection; in the Results Chapter 5, we have noticed a strong correlation between a model and its performance with different input sizes. Thus, it is worth taking into account that with a wider input window size, Whisper works better, Hubert and WavLM meanwhile work better at capturing small dependencies.

Another takeaway that should be considered in future research is the pooling technique. Results of our experiments show that the most effective and universal pooling technique in our setup appeared to be Multihead Attention. Thus, it is definitely worth moving towards attention pooling directions within the question of pooling technique for PII classification with audio features only.

Gold Standard annotation, which was manually done and discussed in Section 4.3 can also be used in future research within the field of PII detection. However, it should be expanded for more reliable results.

In conclusion, our work demonstrates that the task of personally identifiable information (PII) classification using only audio input is not only feasible but can be effectively implemented in both binary and multilabel classification frameworks. Our findings showcase that deep learning models, when appropriately designed and trained, are capable of discerning complex PII cues directly from acoustic features.

As privacy regulations worldwide become increasingly stringent, solu-

tions like ours will be crucial in ensuring that voice-based AI systems operate responsibly and ethically, minimising the risk of accidental data exposure. Ultimately, this work is another step toward empowering organisations to better safeguard sensitive information embedded in spoken communication while supporting the growing demand for trustworthy AI applications.

Bibliography

- Ameer, H., Latif, S., & Fatima, M. (2025). Optimizing multi-stuttered speech classification: Leveraging whisper's encoder for efficient parameter reduction in automated assessment. <https://arxiv.org/abs/2406.05784>
- Asimopoulos, D., Sinosoglou, I., Argyriou, V., Karamitsou, T., Fountoukidis, E., Goudos, S. K., Moscholios, I. D., Psannis, K. E., & Sarigiannidis, P. (2024). Benchmarking advanced text anonymisation methods: A comparative study on novel and traditional approaches. *arXiv preprint arXiv:2404.14465*.
- Baumann, T., Köhn, A., & Hennig, F. (2018). The spoken wikipedia corpus collection. *Language Resources and Evaluation*.
- Canavan, A., Graff, D., & Zipperlen, G. (1997). Callhome american english speech. *Linguistic Data Consortium*.
- Catelli, R., Gargiulo, F., Damiano, E., Esposito, M., & De Pietro, G. (2021). Clinical de-identification using sub-document analysis and electra. *2021 IEEE International Conference on Digital Health (ICDH)*, 266–275.
- Champion, P. (2024). Anonymizing speech: Evaluating and designing speaker anonymization techniques. <https://arxiv.org/abs/2308.04455>
- Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., Yu, X., & Wei, F. (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6), 1505–1518. <https://doi.org/10.1109/jstsp.2022.3188113>
- Cheng, S., Li, Z., Meng, S., Ren, M., Xu, H., Hao, S., Yue, C., & Zhang, F. (2021). Understanding pii leakage in large language models: A systematic survey. *Codex*, 08.
- Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Cohn, I., Laish, I., Beryozkin, G., Li, G., Shafran, I., Szpektor, I., Hartman, T., Hassidim, A., & Matias, Y. (2019a). Audio de-identification: A new entity recognition task. *arXiv preprint arXiv:1903.07037*.

- Cohn, I., Laish, I., Beryozkin, G., Li, G., Shafran, I., Szpektor, I., Hartman, T., Hassidim, A., & Matias, Y. (2019b, June). Audio de-identification - a new entity recognition task. In A. Loukina, M. Morales & R. Kumar (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 2 (industry papers)* (pp. 197–204). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-2025>
- Défossez, A., Copet, J., Synnaeve, G., & Adi, Y. (2022). High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.
- Deu&ser, T., Sparrenberg, L., Berger, A., Hahn&uck, M., Bauckhage, C., & Sifa, R. (2025a). A survey on current trends and recent advances in text anonymization. *arXiv preprint arXiv:2508.21587*.
- Deu&ser, T., Sparrenberg, L., Berger, A., Hahn&uck, M., Bauckhage, C., & Sifa, R. (2025b). A survey on current trends and recent advances in text anonymization. <https://arxiv.org/abs/2508.21587>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. <https://arxiv.org/abs/1810.04805>
- Domingo-Ferrer, J., Sánchez, D., & Soria-Comas, J. (2022). *Database anonymization: Privacy models, data utility, and microaggregation-based inter-model connections*. Springer Nature.
- El Emam, K. (2013). *Guide to the de-identification of personal health information*. CRC Press.
- Gong, Y., Khurana, S., Karlinsky, L., & Glass, J. (2023). Whisper-at: Noise-robust automatic speech recognizers are also strong general audio event taggers. *arXiv preprint arXiv:2307.03183*.
- Graux, H., & Graux, D. M. (2018). Article 29 data protection working party. ec.europa.eu/newsroom/document.cfm.
- Health insurance portability and accountability act of 1996 [URL: <https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf>]. (1996).
- Hong, S., Zou, Y., & Wang, W. (2020). Gated multi-head attention pooling for weakly labelled audio tagging. *Interspeech 2020*.
- Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. <https://arxiv.org/abs/2106.07447>
- Iglesias, M., Del Carmen, M., et al. (2019). *Voice personalization and speaker de-identification in speech processing systems* [Doctoral dissertation, Teoría do sinal e comunicacóns].
- Ji, Z., Shen, Y., Koedinger, K. R., & Lin, J. (2025). Enhancing the de-identification of personally identifiable information in educational data. <https://doi.org/10.5281/ZENODO.17114271>

- Jin, Q., Toth, A. R., Schultz, T., & Black, A. W. (2009). Speaker de-identification via voice transformation. *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, 529–533.
- Kotevski, D. P., Smee, R. I., Field, M., Nemes, Y. N., Broadley, K., & Vajdic, C. M. (2022). Evaluation of an automated presidio anonymisation model for unstructured radiation oncology electronic medical records in an australian setting. *International Journal of Medical Informatics*, 168, 104880.
- Leevy, J. L., Khoshgoftaar, T. M., & Villanustre, F. (2020). Survey on rnn and crf models for de-identification of medical free text. *Journal of Big Data*, 7(1), 73.
- Leygue, T., Sabourin, A., Bolzmacher, C., Bouchigny, S., Anastassova, M., & Pham, Q.-C. (2025). Explainable speech emotion recognition through attentive pooling: Insights from attention-based temporal localization. <https://arxiv.org/abs/2506.15754>
- Li, M., Keizer, S., & Doddipatla, R. (2024). Prompting whisper for qa-driven zero-shot end-to-end spoken language understanding. <https://arxiv.org/abs/2406.15209>
- Lison, P., Pilán, I., Sanchez, D., Batet, M., & Øvrelid, L. (2021a). Anonymisation models for text data: State of the art, challenges and future directions. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4188–4203.
- Lison, P., Pilán, I., Sanchez, D., Batet, M., & Øvrelid, L. (2021b, August). Anonymisation models for text data: State of the art, challenges and future directions. In C. Zong, F. Xia, W. Li & R. Navigli (Eds.), *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 4188–4203). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.acl-long.323>
- Ma, R., Liusie, A., Gales, M. J. F., & Knill, K. M. (2024). Investigating the emergent audio classification ability of asr foundation models. <https://arxiv.org/abs/2311.09363>
- Mallinson, C., Childs, B., & Van Herk, G. (2017). *Data collection in sociolinguistics*. Taylor & Francis.
- Meng, Y., Goldwater, S., & Tang, H. (2025). Effective context in neural speech models. *arXiv preprint arXiv:2505.22487*.
- Miao, X., Wang, X., Cooper, E., Yamagishi, J., & Tomashenko, N. (2022). Language-independent speaker anonymization approach using self-supervised pre-trained models. <https://arxiv.org/abs/2202.13097>
- Miara, V., Lepage, T., & Dehak, R. (2024). Towards supervised performance on speaker verification with self-supervised learning by leveraging

- large-scale asr models. *Interspeech 2024*, 2660–2664. <https://doi.org/10.21437/interspeech.2024-486>
- Moenandar, S.-J., Basten, F., Taran, G., Panagoulia, A., Coughlan, G., & Duarte, J. (2024). The structured narrative interview. *Narrative Inquiry*, 34(2), 307–334.
- Ohm, P. (2009). Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA l. Rev.*, 57, 1701.
- Panariello, M., Nespoli, F., Todisco, M., & Evans, N. (2024). Speaker anonymization using neural audio codec language models. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4725–4729.
- Pang, Y., Yao, L., Xu, J., Wang, Z., & Lee, T.-Y. (2022). Integrating transformer and imbalanced multi-label learning to identify antimicrobial peptides and their functional activities. *Bioinformatics*, 38(24), 5368–5374.
- Patsakis, C., & Lykousas, N. (2023). Man vs the machine: The struggle for effective text anonymisation in the age of large language models.
- Pilán, I., Lison, P., Øvrelid, L., Papadopoulou, A., Sánchez, D., & Batet, M. (2022). The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4), 1053–1101.
- Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. <https://arxiv.org/abs/2212.04356>
- Raj, A., & D’Souza, R. (2021). Anonymization of sensitive data in unstructured documents using nlp. *International Journal of Mechanical Engineering and Technology (IJMET)*, 12(4), 25–35.
- Salazar, J., Liang, D., Nguyen, T. Q., & Kirchhoff, K. (2019). Masked language model scoring. *arXiv preprint arXiv:1910.14659*.
- Sang, E. F., & De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.
- Savkin, M., Ionov, T., & Konovalov, V. (2025, April). SPY: Enhancing privacy with synthetic PII detection dataset. In A. Ebrahimi, S. Haider, E. Liu, S. Haider, M. Leonor Pacheco & S. Wein (Eds.), *Proceedings of the 2025 conference of the nations of the americas chapter of the association for computational linguistics: Human language technologies (volume 4: Student research workshop)* (pp. 236–246). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2025.nacl-srw.23>
- Schuhmann, D. R., Yepes, A. J., van Mulligen, E., Kang, N., Kors, J., Milward, D., Corbett, P., Buyko, E., Tomanek, K., Beisswanger, E., et al. (2010). The calbc silver standard corpus for biomedical named entities—a study in harmonizing the contributions from four in-

- dependent named entity taggers. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.
- Song, Z., Sun, T., Hu, R., Chen, K., & Lu, J. (2025). Leveraging self-supervised learning based speaker diarization for misp 2025 avsd challenge. *Proc. Interspeech 2025*, 1903–1907.
- Sun, Z., Zhang, L., Wang, Q., Zhou, P., & Xie, L. (2025). Towards robust overlapping speech detection: A speaker-aware progressive approach using wavlm. <https://arxiv.org/abs/2505.23207>
- Talmy, S. (2010). Qualitative interviews in applied linguistics: From research instrument to social practice. *Annual review of applied linguistics*, 30, 128–148.
- Tamir, Z., Thebaud, T., Villalba, J., Dehak, N., & Kurland, O. (2025). Multimodal emotion diarization: Frame-wise integration of text and audio representations. *Proc. Interspeech 2025*, 4338–4342.
- Tomashenko, N., Miao, X., Champion, P., Meyer, S., Wang, X., Vincent, E., Panariello, M., Evans, N., Yamagishi, J., & Todisco, M. (2024). The voiceprivacy 2024 challenge evaluation plan. *arXiv preprint arXiv:2404.02677*.
- Tomashenko, N., Wang, X., Vincent, E., Patino, J., Srivastava, B. M. L., Noé, P.-G., Nautsch, A., Evans, N., Yamagishi, J., O'Brien, B., et al. (2022). The voiceprivacy 2020 challenge: Results and findings. *Computer Speech & Language*, 74, 101362.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Voigt, P., & Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide*, 1st Ed., Cham: Springer International Publishing, 10(3152676), 10–5555.
- Wang, W., Wang, W., Sun, M., & Wang, C. (2019). Acoustic scene analysis with multi-head attention networks. *arXiv preprint arXiv:1909.08961*.
- Wu, T., Huang, Q., Liu, Z., Wang, Y., & Lin, D. (2021). Distribution-balanced loss for multi-label classification in long-tailed datasets. <https://arxiv.org/abs/2007.09654>
- Wu, T.-Y., Hsu, T.-Y., Li, C.-A., Lin, T.-H., & Lee, H.-y. (2022). The efficacy of self-supervised speech models for audio representations. *HEAR: Holistic Evaluation of Audio Representations*, 90–110.
- Yao, J., Wang, Q., Guo, P., Ning, Z., & Xie, L. (2024). Distinctive and natural speaker anonymization via singular value transformation-assisted matrix. <https://arxiv.org/abs/2405.10786>
- You, L. L., Pollack, K. T., Long, D. D., & Gopinath, K. (2011). Presidio: A framework for efficient archival data storage. *ACM Transactions on Storage (TOS)*, 7(2), 1–60.

Appendix. Comprehensive Results Overview

Table 1: Comprehensive summary of binary and multilabel PII classification results across models, pooling strategies, context, and granularity. For multilabel, we report mAP, AUC (macro/micro), and F1 (macro/micro). For binary, we report Precision, Recall, and F1.

Granularity	Model	Pooling Strategy	mAP/Precision	AUC (macro)	AUC (micro)	F1 (macro)	F1 (micro)/Recall	F1
<i>Multilabel Classification — Context ($\pm 0.5s$)</i>								
Phrase-level	WavLM	Multihead-Attn	0.33	0.80	0.83	0.35	0.38	
Phrase-level	HuBERT	Multihead-Attn	0.17	0.68	0.70	0.19	0.25	
Phrase-level	Whisper	Multihead-Attn	0.45	0.77	0.91	0.45	0.77	
Word-level	WavLM	Multihead-Attn	0.24	0.91	0.92	0.27	0.37	
Word-level	HuBERT	Multihead-Attn	0.27	0.80	0.95	0.31	0.91	
Word-level	Whisper	Multihead-Attn	0.50	0.99	0.99	0.48	0.53	
Equal-slice (0.5s)	WavLM	Multihead-Attn	0.31	0.86	0.97	0.36	0.94	
Equal-slice (0.5s)	HuBERT	Multihead-Attn	0.27	0.80	0.95	0.31	0.91	
Equal-slice (0.5s)	Whisper	Multihead-Attn	0.19	0.52	0.86	0.20	0.85	
Frame-wise	WavLM	Multihead-Attn	0.56	0.96	0.98	0.56	0.95	
Frame-wise	HuBERT	Multihead-Attn	0.46	0.93	0.95	0.46	0.94	
Frame-wise	Whisper	Multihead-Attn	0.20	0.52	0.75	0.21	0.54	
Phrase-level	WavLM	Two-stage Hier.	0.36	0.82	0.85	0.38	0.46	
Phrase-level	HuBERT	Two-stage Hier.	0.20	0.70	0.75	0.22	0.27	
Phrase-level	Whisper	Two-stage Hier.	0.26	0.76	0.88	0.28	0.62	
Word-level	WavLM	Two-stage Hier.	0.37	0.92	0.94	0.37	0.44	
Word-level	HuBERT	Two-stage Hier.	0.11	0.57	0.61	0.13	0.15	
Word-level	Whisper	Two-stage Hier.	0.20	0.51	0.76	0.09	0.18	
Equal-slice (0.5s)	WavLM	Two-stage Hier.	0.35	0.86	0.98	0.39	0.94	
Equal-slice (0.5s)	HuBERT	Two-stage Hier.	0.25	0.75	0.97	0.28	0.93	
Equal-slice (0.5s)	Whisper	Two-stage Hier.	0.19	0.55	0.83	0.21	0.74	
Frame-wise	WavLM	Two-stage Hier.	0.32	0.85	0.95	0.37	0.93	
Frame-wise	HuBERT	Two-stage Hier.	0.33	0.85	0.96	0.38	0.94	
Frame-wise	Whisper	Two-stage Hier.	0.20	0.53	0.74	0.22	0.67	
Phrase-level	WavLM	Gated (stride 1)	0.34	0.81	0.85	0.30	0.38	
Phrase-level	HuBERT	Gated (stride 1)	0.19	0.67	0.75	0.21	0.23	
Phrase-level	Whisper	Gated (stride 1)	0.24	0.57	0.56	0.25	0.52	
Word-level	WavLM	Gated (stride 1)	0.24	0.92	0.93	0.29	0.35	
Word-level	HuBERT	Gated (stride 1)	0.09	0.80	0.81	0.13	0.13	
Word-level	Whisper	Gated (stride 1)	0.20	0.51	0.69	0.21	0.86	
Equal-slice (0.5s)	WavLM	Gated (stride 1)	0.30	0.84	0.94	0.32	0.90	
Equal-slice (0.5s)	HuBERT	Gated (stride 1)	0.22	0.71	0.88	0.25	0.88	
Equal-slice (0.5s)	Whisper	Gated (stride 1)	0.19	0.51	0.77	0.20	0.69	
Frame-wise	WavLM	Gated (stride 1)	0.32	0.75	0.79	0.36	0.94	
Frame-wise	HuBERT	Gated (stride 1)	0.27	0.78	0.91	0.32	0.93	
Frame-wise	Whisper	Gated (stride 1)	0.20	0.51	0.62	0.21	0.66	
<i>Multilabel Classification — No Context</i>								
Phrase-level	WavLM	Multihead-Attn	0.40	0.85	0.87	0.39	0.47	
Phrase-level	HuBERT	Multihead-Attn	0.20	0.70	0.75	0.22	0.22	
Phrase-level	Whisper	Multihead-Attn	0.44	0.77	0.88	0.43	0.72	
Word-level	WavLM	Multihead-Attn	0.36	0.91	0.93	0.37	0.39	
Word-level	HuBERT	Multihead-Attn	0.23	0.85	0.87	0.20	0.21	
Word-level	Whisper	Multihead-Attn	0.70	0.99	0.99	0.66	0.68	
Equal-slice (0.5s)	WavLM	Multihead-Attn	0.32	0.81	0.96	0.38	0.94	
Equal-slice (0.5s)	HuBERT	Multihead-Attn	0.26	0.73	0.94	0.28	0.90	
Equal-slice (0.5s)	Whisper	Multihead-Attn	0.19	0.51	0.87	0.20	0.82	
Frame-wise	WavLM	Multihead-Attn	0.59	0.96	0.98	0.55	0.95	
Frame-wise	HuBERT	Multihead-Attn	0.47	0.94	0.95	0.45	0.93	
Frame-wise	Whisper	Multihead-Attn	0.20	0.50	0.86	0.21	0.33	

Continued on next page

Granularity	Model	Pooling Strategy	mAP/Precision	AUC (macro)	AUC (micro)	F1 (macro)	F1 (micro)/Recall	F1
Phrase-level	WavLM	Two-stage Hier.	0.41	0.84	0.87	0.40	0.47	
Phrase-level	HuBERT	Two-stage Hier.	0.20	0.69	0.75	0.20	0.22	
Phrase-level	Whisper	Two-stage Hier.	0.23	0.74	0.81	0.27	0.47	
Word-level	WavLM	Two-stage Hier.	0.39	0.91	0.93	0.42	0.43	
Word-level	HuBERT	Two-stage Hier.	0.09	0.54	0.54	0.11	0.15	
Word-level	Whisper	Two-stage Hier.	0.20	0.54	0.78	0.16	0.51	
Equal-slice (0.5s)	WavLM	Two-stage Hier.	0.34	0.83	0.98	0.38	0.95	
Equal-slice (0.5s)	HuBERT	Two-stage Hier.	0.26	0.71	0.96	0.27	0.90	
Equal-slice (0.5s)	Whisper	Two-stage Hier.	0.19	0.54	0.90	0.21	0.80	
Frame-wise	WavLM	Two-stage Hier.	0.31	0.84	0.95	0.36	0.93	
Frame-wise	HuBERT	Two-stage Hier.	0.26	0.77	0.83	0.29	0.94	
Frame-wise	Whisper	Two-stage Hier.	0.20	0.50	0.74	0.21	0.33	
Phrase-level	WavLM	Gated (stride 1)	0.35	0.81	0.85	0.34	0.32	
Phrase-level	HuBERT	Gated (stride 1)	0.21	0.66	0.83	0.22	0.81	
Phrase-level	Whisper	Gated (stride 1)	0.24	0.57	0.51	0.25	0.57	
Word-level	WavLM	Gated (stride 1)	0.31	0.91	0.92	0.28	0.32	
Word-level	HuBERT	Gated (stride 1)	0.11	0.83	0.84	0.15	0.19	
Word-level	Whisper	Gated (stride 1)	0.20	0.51	0.58	0.21	0.74	
Equal-slice (0.5s)	WavLM	Gated (stride 1)	0.29	0.78	0.88	0.32	0.88	
Equal-slice (0.5s)	HuBERT	Gated (stride 1)	0.21	0.66	0.83	0.22	0.81	
Equal-slice (0.5s)	Whisper	Gated (stride 1)	0.19	0.50	0.69	0.20	0.81	
Frame-wise	WavLM	Gated (stride 1)	0.20	0.50	0.03	0.21	0.33	
Frame-wise	HuBERT	Gated (stride 1)	0.20	0.50	0.26	0.21	0.33	
Frame-wise	Whisper	Gated (stride 1)	0.20	0.50	0.03	0.21	0.33	
<i>Binary Classification — Context ($\pm 0.5s$)</i>								
Phrase-level	WavLM	Multihead-Attn	0.62	—	—	—	0.44	0.52
Phrase-level	HuBERT	Multihead-Attn	0.38	—	—	—	0.55	0.45
Phrase-level	Whisper	Multihead-Attn	0.51	—	—	—	0.63	0.56
Word-level	WavLM	Multihead-Attn	0.42	—	—	—	0.44	0.43
Word-level	HuBERT	Multihead-Attn	0.18	—	—	—	0.86	0.30
Word-level	Whisper	Multihead-Attn	0.62	—	—	—	0.64	0.63
Equal-slice (0.5s)	WavLM	Multihead-Attn	0.27	—	—	—	0.40	0.32
Equal-slice (0.5s)	HuBERT	Multihead-Attn	0.18	—	—	—	0.17	0.18
Equal-slice (0.5s)	Whisper	Multihead-Attn	0.03	—	—	—	0.22	0.05
Frame-wise	WavLM	Multihead-Attn	0.18	—	—	—	0.89	0.30
Frame-wise	HuBERT	Multihead-Attn	0.46	—	—	—	0.60	0.52
Frame-wise	Whisper	Multihead-Attn	0.05	—	—	—	0.27	0.09
Phrase-level	WavLM	Two-stage Hier.	0.58	—	—	—	0.51	0.54
Phrase-level	HuBERT	Two-stage Hier.	0.41	—	—	—	0.40	0.41
Phrase-level	Whisper	Two-stage Hier.	0.29	—	—	—	0.38	0.33
Word-level	WavLM	Two-stage Hier.	0.44	—	—	—	0.55	0.49
Word-level	HuBERT	Two-stage Hier.	0.18	—	—	—	0.85	0.30
Word-level	Whisper	Two-stage Hier.	0.06	—	—	—	0.39	0.10
Equal-slice (0.5s)	WavLM	Two-stage Hier.	0.24	—	—	—	0.40	0.30
Equal-slice (0.5s)	HuBERT	Two-stage Hier.	0.13	—	—	—	0.30	0.18
Equal-slice (0.5s)	Whisper	Two-stage Hier.	0.03	—	—	—	0.26	0.06
Frame-wise	WavLM	Two-stage Hier.	0.19	—	—	—	0.89	0.32
Frame-wise	HuBERT	Two-stage Hier.	0.32	—	—	—	0.61	0.42
Frame-wise	Whisper	Two-stage Hier.	0.06	—	—	—	0.28	0.09
Phrase-level	WavLM	Gated (stride 1)	0.39	—	—	—	0.59	0.47
Phrase-level	HuBERT	Gated (stride 1)	0.37	—	—	—	0.49	0.43
Phrase-level	Whisper	Gated (stride 1)	0.29	—	—	—	0.54	0.38
Word-level	WavLM	Gated (stride 1)	0.39	—	—	—	0.59	0.47
Word-level	HuBERT	Gated (stride 1)	0.18	—	—	—	0.85	0.30
Word-level	Whisper	Gated (stride 1)	0.06	—	—	—	0.36	0.10
Equal-slice (0.5s)	WavLM	Gated (stride 1)	0.21	—	—	—	0.47	0.29
Equal-slice (0.5s)	HuBERT	Gated (stride 1)	0.16	—	—	—	0.31	0.21
Equal-slice (0.5s)	Whisper	Gated (stride 1)	0.03	—	—	—	0.60	0.05
Frame-wise	WavLM	Gated (stride 1)	0.25	—	—	—	0.37	0.30
Frame-wise	HuBERT	Gated (stride 1)	0.38	—	—	—	0.55	0.45
Frame-wise	Whisper	Gated (stride 1)	0.06	—	—	—	0.37	0.10

Continued on next page

Granularity	Model	Pooling Strategy	mAP/Precision	AUC (macro)	AUC (micro)	F1 (macro)	F1 (micro)/Recall	F1
<i>Binary Classification — No Context</i>								
Phrase-level	WavLM	Multihead-Attn	0.62	—	—	—	0.56	0.59
Phrase-level	HuBERT	Multihead-Attn	0.41	—	—	—	0.47	0.44
Phrase-level	Whisper	Multihead-Attn	0.60	—	—	—	0.41	0.48
Word-level	WavLM	Multihead-Attn	0.56	—	—	—	0.47	0.51
Word-level	HuBERT	Multihead-Attn	0.18	—	—	—	0.86	0.30
Word-level	Whisper	Multihead-Attn	0.69	—	—	—	0.77	0.73
Equal-slice (0.5s)	WavLM	Multihead-Attn	0.27	—	—	—	0.41	0.33
Equal-slice (0.5s)	HuBERT	Multihead-Attn	0.13	—	—	—	0.30	0.18
Equal-slice (0.5s)	Whisper	Multihead-Attn	0.03	—	—	—	0.21	0.05
Frame-wise	WavLM	Multihead-Attn	0.18	—	—	—	0.88	0.30
Frame-wise	HuBERT	Multihead-Attn	0.37	—	c	—	0.60	0.46
Frame-wise	Whisper	Multihead-Attn	0.05	—	—	—	0.27	0.09
Phrase-level	WavLM	Two-stage Hier.	0.68	—	—	—	0.49	0.57
Phrase-level	HuBERT	Two-stage Hier.	0.44	—	—	—	0.40	0.42
Phrase-level	Whisper	Two-stage Hier.	0.30	—	—	—	0.31	0.31
Word-level	WavLM	Two-stage Hier.	0.53	—	—	—	0.43	0.47
Word-level	HuBERT	Two-stage Hier.	0.18	—	—	—	0.85	0.30
Word-level	Whisper	Two-stage Hier.	0.06	—	—	—	0.16	0.09
Equal-slice (0.5s)	WavLM	Two-stage Hier.	0.31	—	—	—	0.32	0.32
Equal-slice (0.5s)	HuBERT	Two-stage Hier.	0.16	—	—	—	0.25	0.19
Equal-slice (0.5s)	Whisper	Two-stage Hier.	0.03	—	—	—	0.62	0.05
Frame-wise	WavLM	Two-stage Hier.	0.17	—	—	—	0.90	0.28
Frame-wise	HuBERT	Two-stage Hier.	0.41	—	—	—	0.59	0.48
Frame-wise	Whisper	Two-stage Hier.	0.05	—	—	—	1.00	0.09
Phrase-level	WavLM	Gated (stride 1)	0.70	—	—	—	0.48	0.57
Phrase-level	HuBERT	Gated (stride 1)	0.54	—	—	—	0.40	0.46
Phrase-level	Whisper	Gated (stride 1)	0.30	—	—	—	0.58	0.39
Word-level	WavLM	Gated (stride 1)	0.44	—	—	—	0.42	0.43
Word-level	HuBERT	Gated (stride 1)	0.17	—	—	—	0.87	0.29
Word-level	Whisper	Gated (stride 1)	0.06	—	—	—	0.11	0.08
Equal-slice (0.5s)	WavLM	Gated (stride 1)	0.27	—	—	—	0.33	0.30
Equal-slice (0.5s)	HuBERT	Gated (stride 1)	0.15	—	—	—	0.24	0.18
Equal-slice (0.5s)	Whisper	Gated (stride 1)	0.03	—	—	—	0.49	0.05
Frame-wise	WavLM	Gated (stride 1)	0.17	—	—	—	0.90	0.29
Frame-wise	HuBERT	Gated (stride 1)	0.18	—	—	—	0.87	0.29
Frame-wise	Whisper	Gated (stride 1)	0.05	—	—	—	1.00	0.09