Asynchronous Automatic Speech Recognition System

Undergraduate Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Anand Theertha Nakhate
ID No. 2017B3A70660G

Under the supervision of:

Prof. Chng Eng Siong
&
Dr. Pritam Bhattacharya

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, K K BIRLA GOA CAMPUS

December 2021

Declaration of Authorship

I, Anand Theertha NAKHATE, declare that this Undergraduate Thesis titled, 'Asynchronous Automatic Speech Recognition System' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Certificate

This is to certify that the thesis entitled, "*Asynchronous Automatic Speech Recognition System*" and submitted by Anand Theertha NAKHATE ID No. 2017B3A70660G in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Supervisor	*Co-Supervisor*
Prof. Chng Eng SIONG	Dr. Pritam BHATTACHARYA
Assoc. Professor,	Asst. Professor,
Nanyang Technological University, Singapore	BITS-Pilani Goa Campus
Date:	Date:

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, K K BIRLA GOA CAMPUS

Abstract

Bachelor of Engineering (Hons.)

Asynchronous Automatic Speech Recognition System

by Anand Theertha NAKHATE

With the rapid advancement of technology, digital video has become very popular due to its entertainment to the viewer. Billions of new videos and audio data are uploaded every day. Some videos have close captioning, which allows a broad audience to understand the video. To cater to a larger audience, many producers or content developers will include a subtitle or close caption to a larger audience. Some choose to do manual transcription, while others use speech recognition technology to perform automatic transcription. There is much software in the market available that provides automatic transcription services. However, some software can perform both real-time transcription and video/audio transcription while others only either of them.

Furthermore, some of this software can only transcribe the English language, and none allows code-switching. For any group discussions, meetings, or even one-to-one verbal conversations, it is essential to know and note essential conversation segments. To keep note of the crucial segments, a human must take notes of the minutes of the session. This process of note-taking and having important segments noted works but is not efficient and can also lead to loss of information due to human error. Developing an application that can keep track of individual voice data and provide a textual transcript of the speech will make it easier and much more precise to extract important segments from a conversation session. This report will analyze the different technologies and transcribing tools in the market to create a complete automatic speech recognition solution and implement Named Entity Recognition on the transcribed text.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Associate Professor Chng Eng Siong, for providing their invaluable comments, suggestions, and encouragement throughout the course of the project. It has been a fruitful experience and has allowed me to acquire valuable knowledge.

I would also like to express my gratitude toward Mr. Kyaw Zin Tun for his guidance, advice, and patience throughout the project. His valuable advice and patience have aided in the progression of the project. Lastly, I would like to thank all the other people who helped me in various ways to complete the project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The advancement in internet technology and the widespread availability of online video have driven the growth of many networked streaming media applications such as live broadcasts and online education. Young adults frequently visit multiple sites such as Netflix, YouTube, and Hulu, increasing the demand for video streaming media. One of the social networks, YouTube, has more than 2 billion monthly viewers and over a billion hours of videos watched daily. YouTube is localized in over 100 countries and can be accessed in 80 languages. With so many videos across different countries and languages, viewers will come across videos with unfamiliar regional accents of their native language. Viewers may have difficulty understanding the videos due to the speaker's accent mismatching language standards (and/or with the Viewer's accent). The difficulty is magnified when there are many dialects of the same language.

Furthermore, when the Viewer is confronted with a foreign language, the vowels and consonants may fail to match their native sound categories. It will cause the viewers to have difficulty understanding the video. It will cause the viewers to have difficulty understanding the video.

The Speech and Language Research Group at the School Of Computer Science, NTU, has been working in speech research and transcription. The team at NTU has worked on developing various models for audio transcription in ASR, but there is no application that can make it easier for any user to use this ASR. Currently, the application can provide real-time transcription for only two speakers. It also lacks many different features which this project aims to achieve, such as supporting a mixer device, editing the transcript received, etc.

The introduction of automatic speech recognition (ASR) allows the recognition and translation of spoken language into written text. In the current market, there is much software that provides transcription services. However, much software only provides either audio file transcription or real-time transcription. Furthermore, the software in the market does not support multi-language and code-switching. In addition, ASR software also does not guarantee 100% accuracy.

The purpose of this project is to design, develop, and maintain a desktop application that allows for real-time transcription of a group of users in conversation and edit the transcribed audio data to remove any transcription errors. In most call centers, the operator needs to note essential points while talking to the customer. This can be a hectic process in a long day, leading to incorrect data as the worker takes on more calls. With the application, the operator does not need to worry about note-taking. The application will save the audio data for both operator and the customer while the transcription is shown in real-time, which can be saved later in a log file. This makes it easier to log phone calls in a call center. Another area is the meeting room sessions. In any meeting room session, there is a minute taker who keeps notes of all the conversations in the session, this process is entirely dependent on the minute taker, and any information loss by the minute taker will lead to miscommunication between teams in the organization. Using the application in a meeting room session eliminates human errors by a minute taker. The complete meeting session can be projected on the screen, which shows the real-time speech to text transcription for each speaker uniquely identified for other participants to understand easily. The application can record statements given by a person in a court or to the police in any case. It can also be used by community managers or studios where there is a need to record a podcast or interviews. While saving individual speaker's audio, this application will also save the transcription, leading to a hassle-free experience. The application also works as an editor so that the transcripts received from a session can be edited as soon as the session is completed, and no other application is required to edit the transcriptions. The editor provides per transcript audio playback, which is unique to the application and not existent in other editor applications.

Research has been conducted on the performance of popular speech recognition systems such as Microsoft API, Google API, and CMU Sphinx [5]. When using the latest technologies, these systems are only capable of achieving accuracies in about 90%. Therefore, we are still required to do some manual editing to improve the quality of the subtitle. The project focuses on providing a complete ASR solution to address this issue.

1.2 Objectives

- The current implementation of the project need the user to split the audio file using a different application when a user needs to use an audio file with multiple channels. The objective is to create a feature in the application that allows the user to split the audio files in the application.

- Implement a Microphone Audio Level Visualization while selecting a the microphones for a speaker. To provide further help to the user, we create a button that allows the user to record and playback a 5 second audio file.

- Provide a template to display particular rows of the table that is currently being displayed of specific modules.

- Implement a table for Named Entity Recognition that tabulates the details of the tokens and its named entity of the transcript that is currently displayed in the conversation table.

1.3 Project Scope and Assumptions

The scope of this project restricts to providing a complete ASR solution. The ASR solution is limited to real-time transcribing, saving the audio and transcribed file, audio playback, editor, and providing the service of Named Entity Recognition for transcription.

1.4 Report Organizaiton

This project is divided into six chapters and an overview of each chapter is as follows:

- Chapter 1 gives an introduction, objective and scope of this project.

- Chapter 2 discuss the related works of various researchers in named entity recognition and a summary of transcribing tools in the market.

- Chapter 3 will give an overview of the evaluation results of different NER models on some popular and prominent NER datasets.

- Chapter 4 will on focus on reviewing existing work done such as system architecture.

- Chapter 5 discuss the work done and details of the implementation in this project.

- Chapter 6 will conclude the report and some recommendation for future work.

Chapter 2

Literature Review

This chapter aims to provide a comprehensive literature review of research done for Named Entity Recognition. This section will briefly cover the different approaches for solving the problem of NER.

2.1 Approaches to Named Entity Recognition

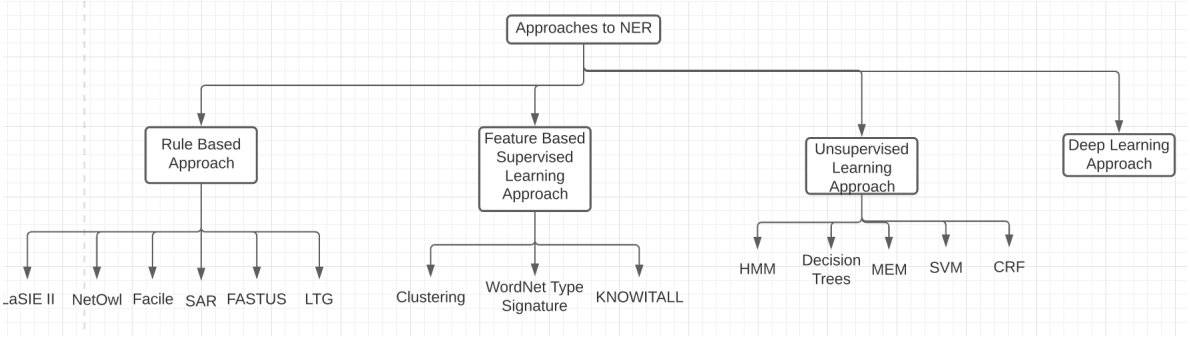

FIGURE 2.1: Approaches to NER

The traditional approaches in solving NER can be divided into three categories: the rule-based approach, the unsupervised learning-based approach, and the feature-based supervised learning approach. [21]. However, deep learning (DL) NER systems have become predominant and obtained state-of-the-art performances over recent times. Compared to the supervised learning methods, neural networks are advantageous in automatically identifying hidden characteristics. We first concisely describe the conventional methods to solving NER and later understand what deep learning is, and then survey DL-based methods to NER.

2.1.1 Rule-Based Approach

What is Rule-Based Approach?

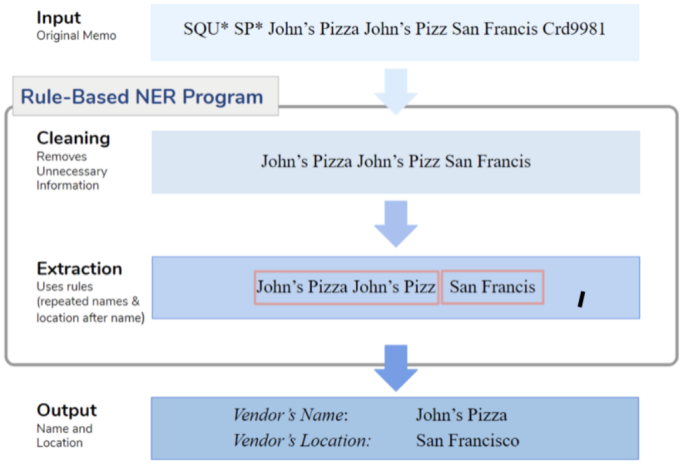

FIGURE 2.2: Rule-Based NER.

Rule-based Named Entity Recognition methods depend on hand-crafted rules. Rules can be devised from domain-specific gazetteers and syntactic-lexical designs. The rule-based systems directly encode the linguistic knowledge of the data set directly into a collection of simple rules [40]. The rule-based NER program generally consists of two parts; cleaning and extraction. In the cleaning phase of the program, we discard unnecessary information from every record to handle the extraction of named entities. In the extraction phase, we recognize the data from the "refined" memo. We identify the unnecessary data for both phases and recognize the patterns of occurrences of named entities for extraction. Rule-based models operate very well when the dictionary is exhaustive [10]. Due to domain-specific rules and fragmentary dictionaries, such systems often witness high precision and low recall, which cannot be conveyed to different domains [40].

What are some of the works done in this approach for NER?

Some of the popular rule-based NER systems that include LaSIE-II in 1998 [39], FASTUS in 1998 [19], SAR in 1998 [10], Facile in 1998 [80], LTG in 1999 [1] and NetOwl in 2005 [27] systems, each of which follows different system architecture and rule patterns for identifying named entity tags.

Some of the later works done in NER using this approach are: Ralph Grishman in 1995 [31] developed a rule-based NER method that employs specialized dictionaries, incorporating the

names of all nations, principal cities, organizations, traditional maiden titles, etc. Kim in 2000 [41] recommended using the Brill rule deducing method for speech data. The system mentioned above produces precepts automatically depending on Brill's POS tagger. In the biomedical field, Hanisch et al. in 2005 [20] proposed a system named ProMiner, which makes use of a pre-processed synonym vocabulary to recognize protein namings and potential genes in biomedical documents. Quimbaya et al. in 2016 [2] suggested a dictionary-using procedure for Named Entity Recognition in computerized medical reports.

What are the advantages of using Rule-Based Approach to solving NER?

While statistical machine learning is popularly applied as a black-box technology concerning traceability and clarity of decisions, rule-based methods support a primarily declarative approach leading to clear and expressive models. Additional advantages of the rule languages include readability, maintainability, and the likelihood of directly conveying domain knowledge into rules. The potential of including the understanding of a field expert into the extraction process instead of adopting the most promising training data, hyper-parameters, or weights, thus indirectly incorporating the domain knowledge, is a notable advantage contrasted to other approaches to IE.

However, rule-based and ML strategies need not be practiced exclusively but can complement one another very well. Supervised learning always needs substantial training data; one of the primary steps is to sample and generate training data sets. In areas where pre-labeled samples are rare or non-existent, this step is the most time-consuming task. To obtain a significant quantity of high-quality training examples, rule languages can be used. This methodology is a form of bootstrapping modern ML technology. [82]

What are the limitations of using Rule-Based approach to solving NER?

Some of the disadvantages of rule-based systems are that The Rule-Based system necessitates a deep understanding of the field and a lot of manual work. Creating rules for a complicated system is very challenging and time-consuming. The system will generate the result as per the rules, so the learning potential of the method by itself is significantly less. If an application we desire to build is too complicated, creating the RB system can take time and investigation. Complex pattern identification is a challenging job in the RB strategy [59].

2.1.2 Unsupervised Learning Based Approach

2.1.2.1 What is Unsupervised Learning?

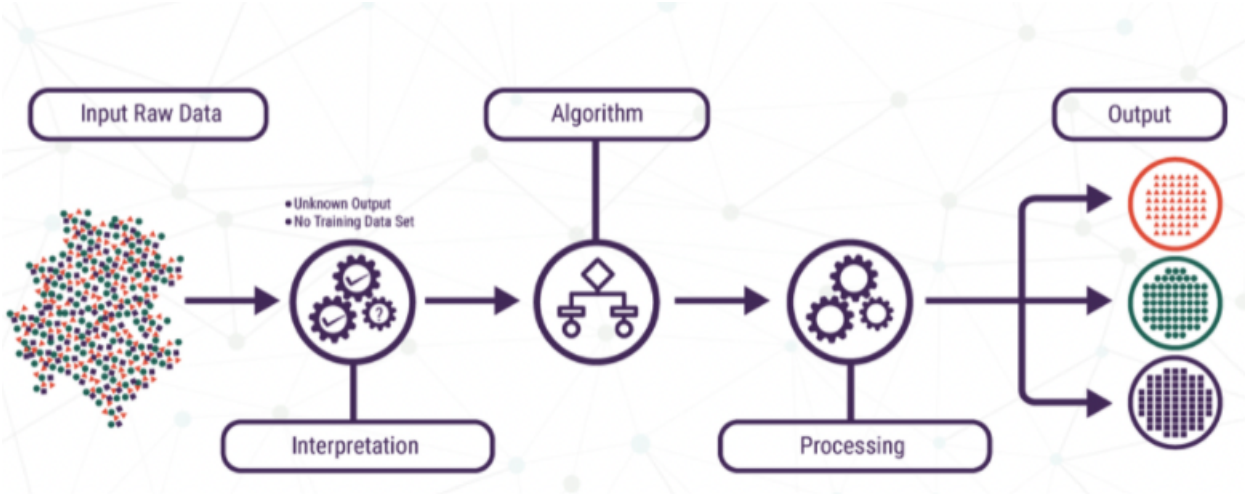

FIGURE 2.3: Unsupervised Learning.

Unsupervised learning refers to applying artificial intelligence algorithms to recognize patterns in the knowledge base comprising data points that are neither classified nor labeled. The main difficulty with supervised learning is the need to name a large number of features. For learning a reliable model, a strong set of features and a large annotated corpus is required. Several languages do not have a large annotated corpus readily available. Unsupervised procedures for NER have been developed to deal with the scarcity of annotated corpus across areas and languages. [73]

2.1.2.2 What are some of the early works done in Unsupervised learning Approaches for NER?

[72] in 1991 and [15] in 1992 are early attempts in this field. The authors introduced the idea to extend ontology with domain-specific data automatically. In [72] in 1991 and [15], authors invented NER systems based on heuristic rules and lexical sources like WordNet [25]. Rau in 1991 [72] implemented a program that extracts organization titles automatically from financial news. The author implemented a program by merging heuristics, exclusion lists, and comprehensive corpus review. Numerous researchers have developed Unsupervised NER methods. Some of the earliest methods required very minimum training data. Collins and Singer in 1999 [16] used only labeled seeds and seven characteristics, including orthography, the context of the

entity, words enclosed within named entities, etc., for classifying and extracting named entities. Alfonesca and Manandhar, in 2002 [5], said that the principal gain of their procedure is its capability to be employed in any language. They implemented an algorithm based on a word-sense disambiguation system. The process employed in their procedure is Aggire's approach [22]. They adopted Aggire's approach because WordNet does not have topic headers, and they applied it to generate them. They investigate the problem of identifying an input term with a relevant NE class. NE classes are obtained from WordNet. The method allots a class signature to every WordNet synset by listing commonly co-occurring terms in a vast corpus. Later, given an input term in a document, the term's context is compared to class signatures and categorized under the most alike class.

Nadeau et al. in 2006 [65] came up with an unsupervised model for gazetteer construction and NER uncertainty determination based on the approaches by O. Etzioni et al. (2005) [24] and Collins and Singer (1999) [16] that combined an extracted gazetteer with commonly available gazetteers to achieve F-scores of 88%, 61%, and 59% on MUC-7 (Chinchor and Robinson, 1997 [11]) location, person, and organization entities, respectively. Nadeau et al. [65] came up with an unsupervised NER method of two modules. The first module is used to generate comprehensive gazetteers of entities, such as a list of cities. The second one employed simple heuristics to recognize and group entities in the context of a given text (i.e., entity disambiguation).

Zhang and Elhadad in 2013 [92] suggested an unsupervised method for obtaining named entities from biomedical documents. Rather than supervised learning methods, their algorithm uses terminologies, statistics from the text, and shallow syntactic knowledge. Tests on two mainstream biomedical datasets exhibit the efficacy and generalizability of their unsupervised method.

2.1.2.3 What are some of the most prominent systems in Unsupervised NER?

One of the most prominent systems for NER is KNOWITALL. Much literature in IE has been sone on tiny datasets utilizing hand-label training samples. Manual training samples have allowed statistical tools such as Hidden Markov Models or Conditional Random Fields to obtain data from complicated expressions. In distinction, KNOWITALL focuses on unsupervised IE methods from the Web. KNOWITALL takes a collection of predicate signatures as input but no manually annotated training samples of any type and bootstraps its extraction method from a small collection of generic extraction patterns. KNOWITALL utilizes a unique generate-and-test

design to achieve high precision, relying on mutual-information statistics calculated over the Web corpus.

In O. Etzioni et al. (2005) [24], Pointwise Mutual Information and Information Retrieval (PMI-IR) is applied as a trait to evaluate that an NE could be categorized following a given class. PMI-IR, introduced by P. Turney (2001) [75], estimates the dependency among two phrases utilizing web queries. A huge PMI-IR suggests that phrases favor co-occurring. O. Etzioni et al. [24] formulate characteristics for each class (e.g., London) and a vast amount of instinctively created discriminator phrases. KNOWITALL is a domain-independent system developed by O. Etzioni et al. (2005) [24] that extracts data from the Web in an unsupervised, open-ended fashion. KNOWITALL employs eight domain-independent extraction models to produce candidate events. It later automatically examines the plausibility of the candidate facts using PMI calculated using large web text as corpus. Depending on PMI score, KNOWITALL affiliates a probability with every fact it extracts, permitting it to control the trade-off between precision and recall. It relies on a bootstrapping procedure that induces seeds from generic extraction patterns and automatically produces discriminator expressions.

2.1.3 Feature Based Supervised Learning Approach

2.1.3.1 What is Supervised learning for NER?

.50.6.5

FIGURE 2.4: Supervised Learning.

Supervised learning approaches are a class of algorithms that learn a pattern by studying annotated training instances [88]. Using supervised learning, NER is projected to a multi-class classification or sequence labeling task. Features are thoughtfully outlined for some annotated data samples to represent each training sample. ML models are then employed to study a model to identify related patterns from test data. Feature engineering is crucial in supervised NER systems. The characteristic vector is a notion over text where a term is described by one or more Boolean, numeric, or nominal values. Word-level characteristics list lookup traits, and document and corpus characteristics have been broadly used in several supervised NER models. Depending on those features, several noteworthy machine learning models have been implemented in supervised NER, including Hidden Markov Models (HMM), Decision Trees, Maximum Entropy Models, Support Vector Machines (SVM), and Conditional Random Fields (CRF). [48] Typically,

supervised methods either learn disambiguation rules based on discriminative characteristics or learn the assumed distribution parameter that maximizes the likelihood of training data [88].

2.1.3.2 What are some of the popular models for Supervised NER?

Hidden Markov Model (HMM)

HMM is the initial model employed for solving the NER problem by Bikel et al. in 1999 [8]. Bikel proposed a system, IdentiFinder, to identify NER. Based on Bikel's formulation of the problem in the Identifinder model, only an individual label can be attributed to a word in context. Consequently, the system attributes either one of the desired classes to each term or the label NOT-A-NAME to express "none of the desired classes." The HMM-based chunk tagger yielded an accuracy of 96.6% on the MUC-6 dataset and 94.1% on the MUC-7 dataset. Zhou and Su in 2002 [94] used HMM, a NER system on MUC-6 and MUC-7 data, achieving 96.6% and 94.1% F scores, respectively. They included 11 orthographic features, a list of trigger words for the named entities, and a list of words (10000 for the person entity class) from various gazetteers.

Maximum Entropy Model (MEM)

The maximum entropy model, unlike HMM, is a discriminative model. Presented with a set of features and training data, the system directly learns the weight for discriminative features for classification. In Maximum entropy models, the purpose is to maximize the entropy of the data to generalize as much as possible for the training data. The MENE system from Borthwick in 1999 [9] employs a distinct set of information sources in producing its tagging judgments. It uses a broad collection of gazetteers and dictionaries of individual or multi-word expressions like first name, organization name, corporate suffixes. It applies a wide diversity of features like binary features, lexical features, section features, external systems output, consistency, and reference resolution. Curran and Clark in 2003 [18] employed the maximum entropy model to the NER problem. They used the softmax strategy to formulate the probability P(y—x). Curran reported the accuracies of 84.89% for the English test dataset and 68.48% for the German test dataset of the CoNLL-2003 shared task. Malouf in 2002 [58] compared the HMM with Maximum Entropy (ME) by combining multiple features. Their most robust system incorporated capitalization, whether a word was the first in a sentence, a word had appeared before with a known last name, and 13281 first names were collected from various dictionaries. The model achieved a 73.66% and 68.08% F-score on the Spanish and Dutch CoNLL 2002 datasets.

Support Vector Machines (SVM)

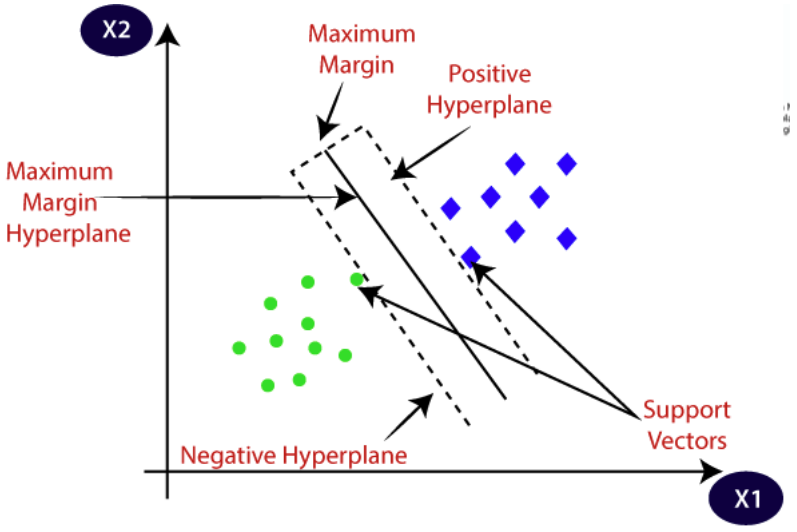

FIGURE 2.5: SVM.

Support Vector Machine was first introduced by Cortes and Vapnik (1995) [17] based on learning a linear hyperplane that divides positive samples from negative samples by a vast margin. The vast margin implies that the gap between the hyperplane and the point from either instance is maximum. The points closest to the hyperplane on both sides are known as support vectors. McNamee and Mayfield (2002) [61] tackle the problem as a binary decision problem,i.e., if the term refers to one of the eight classes, i.e., B- Beginning, I- Inside tag for the person, organization, location, and misc tags. Thus there are eight classifiers trained for this purpose. All features used were binary. For the CoNLL 2002 dataset, published accuracies were 60.97 and 59.52 for Spanish and Dutch, respectively. Li et al. (2005) [51] implemented an SVM model on the CoNLL 2003 data and CMU seminar papers. They tested with many window sizes, features from neighboring words, weighting neighboring word features based on their position, and class weights to adjust positive and negative class. They employed two SVM classifiers, one for identifying named entity starts and one for identifying ends. They scored 88.3% F score on the English CoNLL 2003 data.

Conditional Random Field (CRF)

Lafferty et al. (2001) [44] introduced the conditional random field as a statistical modeling mechanism for pattern identification and machine learning using structured prediction. McCallum and Li (2003) [60] proposed a feature induction system for CRF in NE. Tests were conducted on CoNLL 2003 shared task data and obtained an accuracy of 84.04% for English and 68.11% for German datasets. In DrugNER, Liu et al. (2015) [52] obtained state-of-the-art results by

adopting a CRF with features like lexicon resources from Food and Drug Administration (FDA), DrugBank, Jochem , and word embeddings (trained on a MedLine corpus). For the same task, Rockta schel et al. (2013)[77] applied a CRF with features built from dictionaries, ontologies (ChEBI ontologies), prefixes-suffixes from chemical entities, etc.

2.1.4 Deep Learning Based Approaches

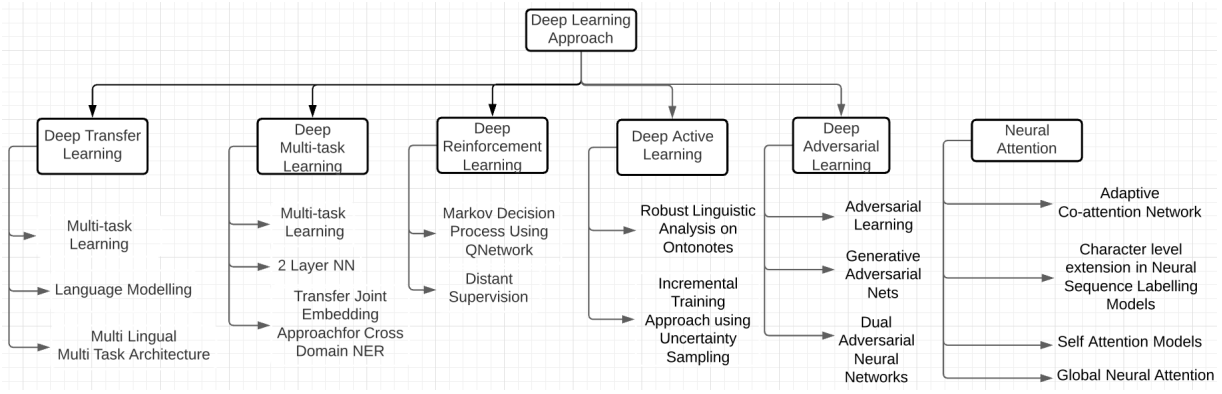

FIGURE 2.6: Deep Learning Approaches to NER

2.1.4.1 What is deep Learning NER?

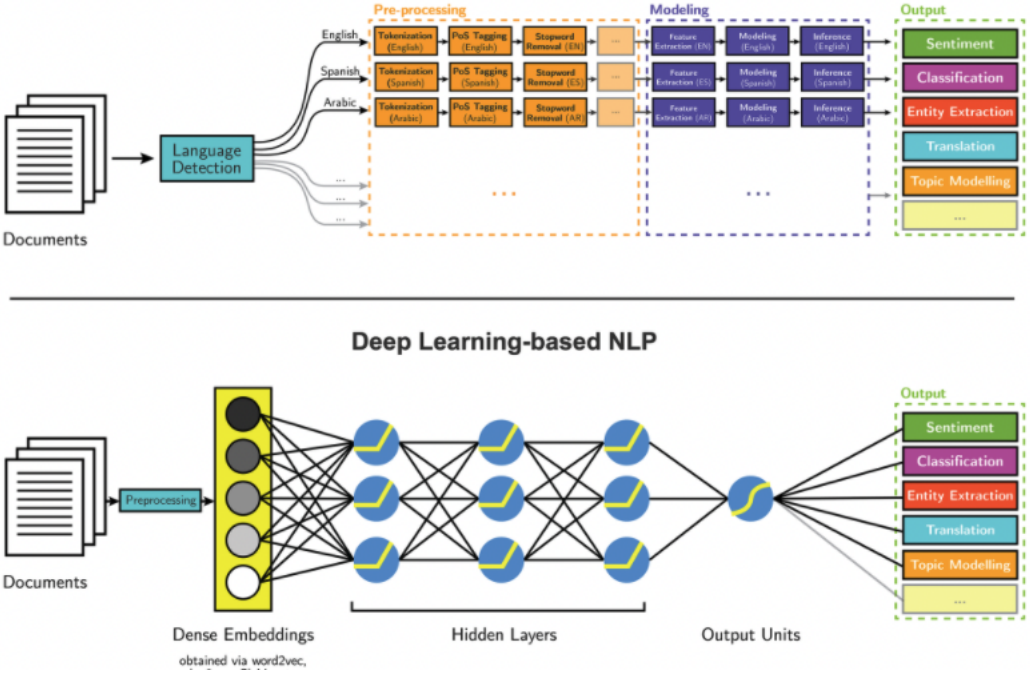

FIGURE 2.7: Deep Learning.

Recently deep learning NER practices have grown predominant and obtained state-of-the-art outcomes. Unlike supervised methods, DL is advantageous in automatically recognizing hidden characteristics. We first briefly present DL, including why DL is for NER. We later inspect DL-based NER strategies.

Deep learning, a refined framework of ML, utilizes multi-layer artificial neural networks to study tasks. This Neural Network (NN) processes input data by many layers of nonlinear transformations to estimate a final result. The layers close to the data study simple traits, while the deeper layers determine complicated characteristics emerging from the lower layers. A feedforward Neural Network (NN) contains an input layer, several hidden layers, and a result layer. The hidden layers' nodes use the previous layer's output and pass outcomes to the subsequent layer.

2.1.4.2 What are the advantages of Deep Learning-based NER?

There are three keen advantages of employing DL procedures to Named Entity Recognition. First, Named Entity Recognition profits from the nonlinear transformation, producing nonlinear mappings from data to outcome. Unlike the linear models (e.g., log-linear HMM), DL-based systems can study complicated and intricate characteristics from data through nonlinear activation functions. Next, DL saves notable work on outlining NER features. The conventional feature-based strategies need a large quantity of engineering talent and field expertise. DL-based systems automatically determine valuable representations and underlying features from raw input data. Third, gradient descent can train DNN NER systems in an end-to-end model. This attribute allows us to produce likely complicated NER practices. [48]

2.1.4.3 What are some of the different architectural systems for Deep Learning NER?

DL models for NER are mainly categorized into Convolution Neural Networks (CNN) [87], Recurrent Neural Networks (RNN) [23] and Transformer architectures. Even though RNN is competent in controlling far-distant dependencies, they seldom fail due to fading gradient problems [[7], [71]]. This led to the advancement of more complex models of RNN: Long Short-Term Memory (LSTM) [35], Bidirectional long short-term memory (BI-LSTM) [30], and Gated Recurrent Unit (GRU) [14].

The use of DNN for NER was established by the CNN-CRF system [70], where CNN is employed to obtain word-level knowledge of the input token embeddings. Then, a CRF layer is superimposed on its top as a tag decoder. By implementing a fixed-sized contextual window, the system obtained an 0.8959 F1 score in the Named Entity Recognition task applying word embeddings [70] and external training data.

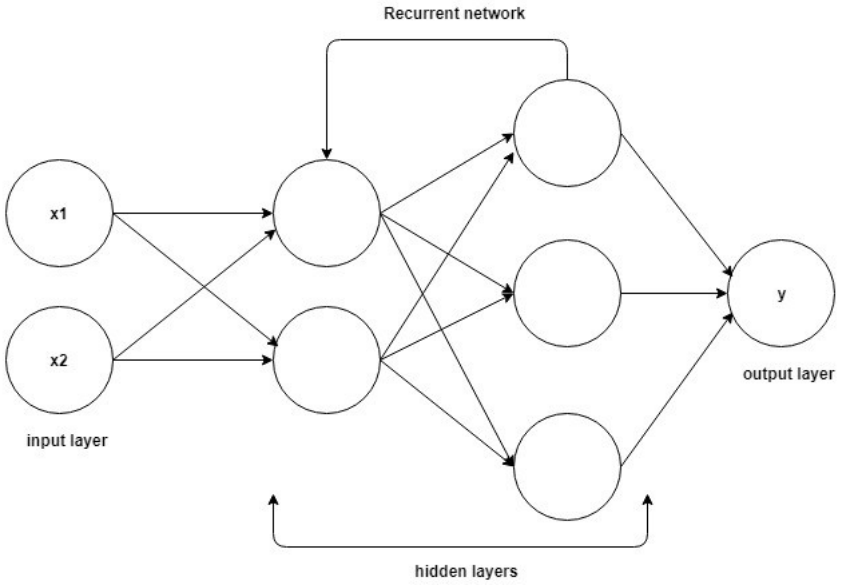

FIGURE 2.8: Recurrent Neural Network.

Huang et al. [90] suggested a variety of RNN-based designs for the NER. These systems are the LSTM system, BI-LSTM system, LSTM-CRF, and BI-LSTM- CRF. In BI-LSTM-CRF, word embeddings and extra word characteristics were provided into the BI-LSTM to produce the word-level design. This design was then conveyed to the outcome CRF layer to predict result annotations. The BI-LSTM-CRF method is less dependent on word embeddings as opposed to the CNN-CRF model [70] and had obtained an F1 score of 90.10% utilizing both Senna embeddings (the same as in [70]) and gazetteer traits. Chiu and Nichols [3] applied CNNs to excerpt character traits for every token from character embedding and character-type traits. The character vectors are concatenated with the word embeddings and extra word features for every word. Then, this concatenated data is provided to multiple layers of the LSTM network that are attached in sequence. The system utilized openly available word embeddings trained by Collobert et al. [70]. The system trained on both training and validation datasets of the CoNLL-2003 dataset [74] obtained an F1 score of 0.916.

Lample et al. [28] introduced a structure comparable to [12] but used the BI- LSTM layer to obtain the character-level traits. The character embeddings for every character in a term are

supplied to a forward and backward LSTM in direct and reversed order. The character-level information is concatenated with pre-trained word embeddings, generated using skip-n-gram [81]. During the training stage, the fine-tuning of word embeddings produced an F Score of 90.94%. Ma and Hovy [57] introduced a NN design of BI-LSTM, CNN, and CRF that could automatically extract the word and character-level traits by utilizing GloVe 100-dimensional word embeddings [68]. The design obtained a top result of 91.21% F1 without utilizing any character-type traits [12].

Yang et al. [91] proposed a hierarchical NER design by employing deep GRUs on both character and word levels, accompanied by a CRF layer to obtain the output tag. The GRUs captured the morphological knowledge at the character-level, and n-gram designs, semantics on the word level utilizing the word embeddings trained by [70]. The fine-tuning of word embeddings enabled the model to achieve a 91.20% F1 score through the training stage.

Neural sequence labeling practices are usually dependent on complicated convolutional or recurrent systems containing encoders and decoders. Transformer, introduced by Vaswani et al. [128], solely controls recurrence and convolutions. A transformer employs stacked self-attention and point-wise, fully connected layers to generate fundamental blocks for encoder and decoder. Tests on numerous tasks [4], [67], [64] display transformers to be more reliable in quality when needing a significantly shorter time to train.

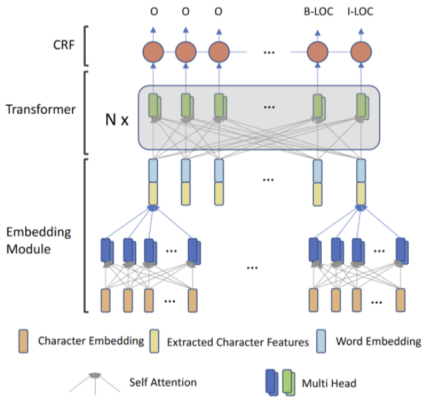

FIGURE 2.9: Transformers.

Based on Transformer, Radford et al. [3] proposed a generic pre-trained transformer (GPT) for language comprehension tasks. GPT holds a two-stage training process. First, they employ a language modeling goal with Transformers on unlabeled information to define the initial parameters. Then they accustom these parameters to an objective task utilizing the supervised goal, leading to minimal modifications to the pre-trained system. Unlike GPT (a left-to-right

architecture), Bidirectional Encoder Representations from Transformers (BERT) is aimed to pre-train deep bidirectional Transformer by simultaneously conditioning on both left and right context in all layers [36].

These language model embeddings pre-trained using Transformer are shifting a new standard of NER. First, these embeddings are contextualized and could substitute traditional embeddings, such as Google Word2vec and Stanford GloVe. Some researches [[53], [38], [86], [56], [54], [37]] have presented convincing performance via using the blend of traditional embeddings and language model embeddings. Next, these language model embeddings can be fine-tuned with an additional output layer for several tasks, including NER.. Especially, Li et al. [49], [50] expressed the NER task as a machine reading comprehension (MRC) problem that could be solved by fine-tuning the BERT system.

Comparison between different Architectures

Primarily, no agreement has been reached about whether external knowledge should be combined into DL-based NER systems. Some studies [53], [38], [86], [95] show that NER performance can be increased with external knowledge. Nevertheless, the limitations are also self-evident in obtaining external knowledge (labor-intensive and computationally costly); and integrating external knowledge negatively influences end-to-end learning and damages the generality of DL-based systems. Second, the Transformer encoder is more potent than LSTM when the Transformer is pre-trained on a massive corpora. Transformers fail on NER jobs if they are not pre-trained and have inadequate training data [32], [33]. On the other hand, the Transformer encoder is faster than recursive layers when the length of the sequence n is smaller than the dimensionality of the representation [4]. Third, a major limitation of RNN and Pointer Network decoders rests in greedy decoding, which indicates that the input of the current level needs the output of the preceding level. This tool may have a notable influence on the speed and impedes parallelization.

For end-users, what architecture to employ is data and domain task-dependent. If data is ample, training models with RNNs from scratch and fine-tuning contextualized language models could be considered. If information is rare, using transfer approaches might be a more suitable choice. [48]

2.2 Annotations used for Named Entity Recognition

Various annotation designs have been applied to different datasets over the years. However, choosing the ideal annotation scheme is a complex problem. We will discuss some of the annotation schemes that are commonly used in the literature. These schemes are the following: [6]

IO: The IO scheme of annotation is the most straightforward scheme applied to Named Entity Recognition. Each token from the dataset is attached to one of two tags in this annotation. One is an inside tag (I) and an outside tag (O). The Inside tag is for the named entities, whereas the O tag is for ordinary words. This annotation has a flaw, as it cannot perfectly encode connected entities of the identical type. The annotation was used by Mozharova and Loukachevitch [63] to annote Russian Text to analyse the impact of IO and IOB annotation schemes for NER.

IOB: The IOB design is also noted in the literature as the BIO scheme, and it has been affirmed by the Conference on Computational Natural Language Learning (CoNLL)[74]. The IOB format is standard for tagging terms in a chunking task in Computational Linguistics such as the Named Entity Recognition task. It designates a tag to each term in the text, deciding whether it is the beginning (B) of a known named entity, the inside (I) of it, or outside (O) of any known named entities.

IOE: The IOE scheme of annotation operates almost similarly to IOB, though it indicates the end of the named entity (E tag) rather than its start.

IOBES: An alternative to the IOB system is IOBES, which enhances the amount of knowledge related to the boundaries of named entities. Apart from tagging the words at the beginning (B), the inside (I), the end (E), and the outside (O) of a named entity. It also identifies single-token entities with the tag S.

BI: The BI scheme of tagging annotates the entities in a similar manner to IOB. Additionally, it marks the beginning of non-entity terms with the tag B-O and the others as I-O.

IE: This system works almost like the IOE annotation with the difference that it annotates the end of non-entity terms with the tag E-O and the rest as I-O.

BIES: The BIES scheme of annotation encodes the entities similar to the IOBES system of annotation. Additionally, it also annotates the non-entity words employing the same method. It

applies B-O to tag the beginning of non-entity words, I-O to tag the inside of non-entity words, and S-O for single non-entity tokens that exist between two entities.

Though various annotation paradigms have been mentioned in the literature, determining the ideal annotation scheme for a particular system is a complicated problem, and it needs to be explored based on the language and text structure and several works have been done in this area. [43]

2.3 Evaluation of an NER System

NER models are generally assessed by examining their outputs upon human annotations. The Examination can result in either an exact match or a relaxed match. Named Entity Recognition typically comprises two subtasks: boundary detection and entity class determination.

In the exact-match evaluation [74], a correctly identified occurrence needs a model to precisely recognize its boundary and class. More particularly, the numbers of False positives (FP), False negatives (FN), and True positives (TP) are applied to compute the metrics such as Precision, Recall, and F-score.

False Positive (FP) refers to an entity identified by a NER model but not present in the ground truth. False Negative (FN) refers to the entity not identified by a NER model but present in the ground truth. True Positive (TP): refers to the entity identified by a NER model and present in the ground truth.

Two primary measures for examining the performance of any information extraction method are precision and recall measures. The precision metric is the proportion of the system's outcomes that are correctly identified. The Recall metric is the proportion of the total entities precisely identified by the NER model.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

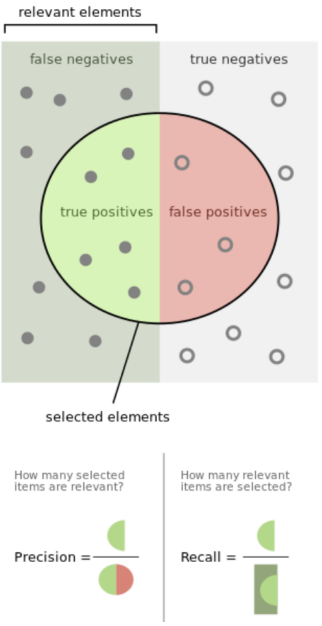

FIGURE 2.10: Evaluation of NER.

A metric that couples both the above metrics is the harmonic mean of precision and recall, the classic F- measure or balanced F-score:

$$F - score = 2 * \frac{(Precision + Recall)}{(Precision * Recall)}$$

Additionally, the macro-averaged F-score and micro- averaged F-score analyze the performance across multiple entity classes. Macro-averaged F-score individually measures the F-score on various entity types, then takes the mean of the F-scores. Micro-averaged F-score totals the individual false positives, false negatives, and true positives overall entity types and then applies them to prepare the statistics. The latter can be massively influenced by the quality of identifying entities in large classes in the text.

An annotation is considered to be a relaxed match when a correct type is considered if an entity is assigned its correct class indifferent of its borders provided that there is an overlap with ground truth borders; a true boundary is considered irrespective of the entity's class assignment. Various complex evaluation methods have been used in various works such as ACE2005 [26] that are not natural and complicate error analysis. Therefore, complex evaluation techniques are not been popularly adopted in recent studies. [48]

2.4 Transcribing Tools

Many transcribing tools in the market use ASR technology to automatically convert real-time speech or audio files into text. With ASR, the user is still required to do manual editing as ARS does not guarantee high accuracy. However, these tools often only convert real-time speech or audio files to text but do not provide audio playback that is needed for manual editing of the transcribed text. Furthermore, some tools only support the English language, while others only support a limited language. Most transcribing tools do not support mixed languages such as Mandarin-English or Malay-English speech.

Transcribing tools also provide only an editor for manual transcribing.

2.4.1 Descript

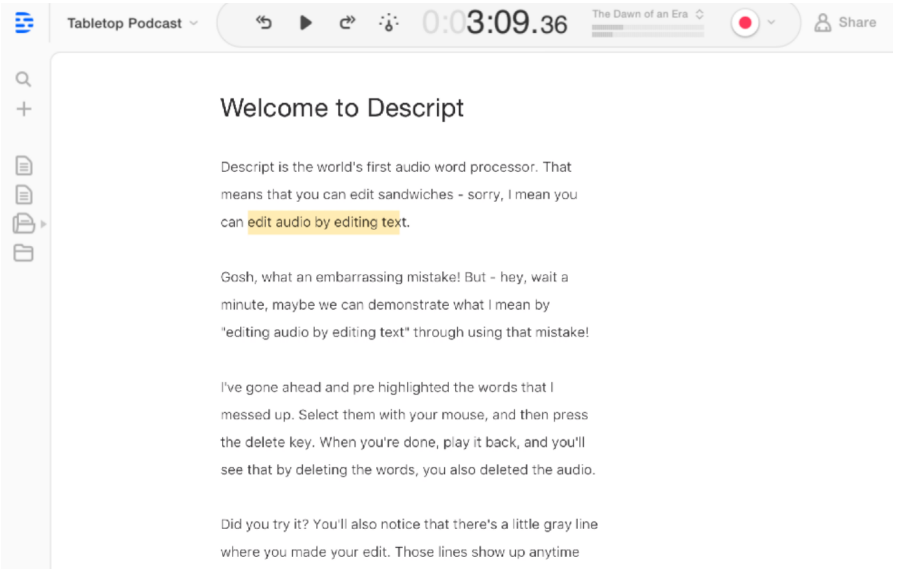

FIGURE 2.11: Descript

Descript is a collaborative video/audio editor that works like Google Docs. It can transcribe video/audio in a short time. Besides transcription, it includes a computer screen recording, publishing, and complete multitrack editing. When transcribing video/audio, Descript has an advanced feature that allows automatic speaker detection. Descript will play a small clip of each speak- er's voice when the speaker is identified for the user to add the names. The speaker labels will update automatically appear in your transcript, and the speaker names will also be color-coded. Similar to Trint, Descript also allows ease of collaboration. User can share their project with a web link and grant collaborators permission to access, comment, or edit [13].

2.4.2 Otter

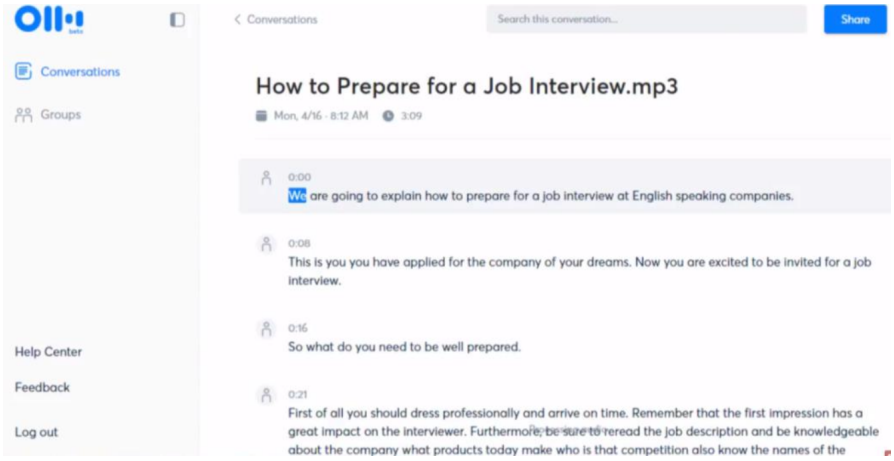

FIGURE 2.12: SVM.

Otter is an automatic Speech-to-Text transcription application developed by a Cali- fornia-based technology company. Otter speech transcription technology is trained with deep learning algorithms using millions of hours of audio . Otter can be accessed anytime and anywhere as it is available in android, iOS, and web browser. Otter can perform live transcription of meetings, interview, and conversation. The user can record conversation through their phone or web browser and the transcrip- tion will be automatically generated. Otter allows user to edit, search and share the recording and transcript. This allows multiple users to view and edit the transcript at the same time which can improve the overall productivity when working in a group. Otter also has integration with other vendors such as Zoom and Dropbox. Otter is very useful and offers many benefits, however, Otter only support English and re- gional accents

Chapter 3

Experimental Results

In this chapter, we try to briefly describe some of the popular and prominent datasets for Named Entity Recognition task. Later on, we evaluate some of the state-of-the-art NER models by training/using a pretrained model to estimate the F1 scores on the said datasets.

3.1 Datasets

The following are some of the prominent and popular datasets that are publicly available for NER task.

3.1.1 CoNLL-2003

CoNLL 2003 [74] dataset was introduced by Sang et al. in Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. CoNLL-2003 is a NER dataset published as a part of CoNLL-2003 shared task: language-independent named entity recognition. The dataset consists of eight records over two languages: English and German. For each language, there is a training file, a development file, a test file, and a large file with unannotated data. Every word in the dataset has been placed on a separate line, and there is a blank line following each sentence.

The first object on every line is a word, following that is its part-of-speech (POS) tag, the third item is a syntactic chunk tag, and the last column denotes its named entity tag. The named entity labels have the form I-TYPE, indicating that the word is inside an expression of type

TYPE. If two phrases of the identical class immediately occur, the first word of the latter phrase will have the tag B-TYPE to record that it starts a different phrase. A word with the tag O is not part of a phrase. train.txt, valid.txt, and test.txt in the dataset have sentences accompanying with their tags. We need only the named entity tags. We extract the words along with their named entities into an array. This tagging scheme is the IOB scheme.

TABLE 3.1: CoNLL-2003 Dataset Description

English Dataset	Articles	Sentences	Tokena	LOC	MISC	ORG	PER
Training Set	946	14987	203621	7140	3438	6321	6600
Development Set	216	3466	51362	1837	922	1341	1842
Test Set	231	3684	46435	1668	702	1661	1617

3.1.2 Ontonotes5

OntoNotes Release 5.0 [85] is the last release of the OntoNotes project, a collaborative work between BBN Technologies, the University of Colorado, the University of Pennsylvania, and the University of Southern California's Information Sciences Institute. The project's aimed to annotate a large corpus containing multiple styles of text (news, conversational telephone speech, weblogs, newsgroups, broadcast, talk shows) in three different languages (English, Chinese, and Arabic) with structural information (syntax and predicate-argument structure) and shallow semantics.

This publication contains 2.9 million words with counts displayed in the following table.

TABLE 3.2: Word Counts in Ontonotes 5.0 Dataset publication

Ontonotes5.0	Arabic	English	Chineese
News	300k	625k	250k
BN		200k	250k
BC		200k	150k
Web		300k	150k
Tele		120k	100k
Pivot			300

The Named entity types are PERSON, NORP, FACILITY, ORG, GPE, LOCATION, PRODUCT, EVENT, WORK OF ART, LAW, LANGUAGE, etc. (18 entities) This tagging scheme uses markup tags using angular brackets for representing named entity $\langle ENAMEXTYPE = "ORG" \rangle Disney \langle /ENAMEX \rangle isaglobalbrand.$

3.1.3 BC5CDR

Built by Li et al. at 2015 [47], the BC5CDR Disease (BC5-Disease) Dataset contains three different collections of articles with chemicals and their relationships annotated, in the English language. BC5CDR corpus contains 1500 PubMed articles with 4409 annotated chemicals, 5818 diseases and 3116 chemical-disease interactions. BC5CDR is a popular biomedical NER dataset.

3.2 Evaluauation of NER Systems

3.2.1 CONLL 2003 Dataset

3.2.1.1 LUKE

LUKE (Language Understanding with Knowledge-based Embeddings) [89] is a pre-trained contextualized design of words and entities based on the transformer. It was presented in the research LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. It attains state-of-the-art outcomes on major NLP benchmarks, like CoNLL-2003 (named entity recognition) and Open Entity (entity typing).

LUKE model proposes unique pre-trained contextualized descriptions of words and entities based on the bidirectional transformer[79]. The model handles words and entities in a text as independent tokens and outputs contextualized descriptions of them. The model is trained employing a new pretraining task dependent on the masked language model of BERT [55]. The task includes predicting randomly masked expressions and entities in a sizeable entity-annotated corpus reclaimed from Wikipedia. The model also introduces an entity-aware self-attention tool that extends the self-attention tool of the transformer and recognizes the types of tokens (words or entities) when estimating attention scores. The model produces impressive experimental state-of-the-art results on Open Entity (entity typing) and CoNLL-2003 (named entity recognition)

The above figure displays the design of LUKE. The system uses a multi-layer bidirectional transformer [79]. It treats words and entities in the text as input tokens and calculates a representation for every token. Formally, given a series consisting of m terms w1, w2, ..., wm and n entities e1, e2, ..., en, the model calculates D-dimensional word representations and entity representations. The entities can be Wikipedia entities or special entities.

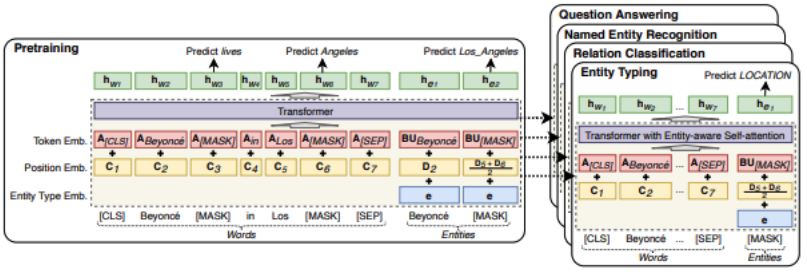

FIGURE 3.1: LUKE Architecture.

Entity Aware Self Attention Mechanism

The self-attention mechanism is the basis of the transformer vaswani2017attention and associates tokens with each other dependent on the attention points among each pair of tokens. For a sequence of input vectors x1, x2, ..., xk, each of the output vectors y1, y2, ..., yk, is calculated based on the weighted aggregate of the transformed input vectors. Each input and output vector resembles to a token (a word or an entity) in the model.

Since LUKE manages two kinds of tokens (i.e., words and entities), the model assumes that it is helpful to utilize the knowledge of target token types when calculating the attention scores. The model then enhances the mechanism by introducing an entity-aware query mechanism that utilizes a separate query matrix for every possible pair of token types.

The computational expenses of the original tool and the stated tool are the same except for the added cost of calculating gradients and updating the parameters of the added query matrices at the training stage.

TABLE 3.3: Evaluation of LUKE NER model on CONLL 2003 Dataset

LUKE CONLL 2003	Precision	Recall	F1 Score	Support
LOC	0.9558	0.9478	0.9518	1666
MISC	0.8553	0.8688	0.862	701
ORG	0.9287	0.9496	0.9391	1647
PERSON	0.9683	0.9719	0.9701	1602
micro-average	0.9386	0.9453	0.942	5616
macro-average	0.927	0.9345	0.9307	5616
weighted-average	0.9389	0.9453	0.9421	5616

TABLE 3.4: Evaluation of LUKE NER model on Open Entity Dataset

LUKE Open Entity	Precision	Recall	F1-Score
	0.798	0.7657	0.7816

3.2.1.2 T-NER

The model, T-NER [78] that stands for transformer-based Named Entity Recognition, is a Python library for NER LM finetuning. An extension to its practical use, T-NER aids the study and examination of the cross-domain and cross-lingual generalization capability of LMs finetuned on NER. This paper shows the library's potential by compiling nine public NER datasets into a consolidated format and estimating the cross-domain and cross-lingual results across the datasets. The outcomes from the initial analyses reveal that in-domain outcomes are usually competitive across datasets. Nevertheless, cross-domain generalization is challenging even with a large pre-trained LM, which can learn domain-specific peculiarities if finetuned on a consolidated dataset. An essential aim of this model was to build a self-contained, comprehensive arrangement to train, evaluate, and efficiently utilize NER models, not only for research objectives but also practical use cases in the industry.

TABLE 3.5: Evaluation Results of T-NER on CONLL-2003 Dataset

T-NER	Precision	Recall	F1 Score	Support
LOC	0.91	0.83	0.87	1660
MISC	0.75	0.7	0.72	702
ORG	0.83	0.89	0.86	1661
PERSON	0.95	0.95	0.95	1610
micro-average	0.88	0.87	0.87	5633
macro-average	0.86	0.84	0.85	5633
weighted-average	0.88	0.87	0.87	5633

3.2.1.3 Flair Model + Cross-Weigh

Eugene Siow [76] trains a flair model using stacked embeddings (with word and flair contextual embeddings) to perform named entity recognition (NER). The dataset used is the CoNLL 2003 dataset for NER (train, dev) with a manually corrected (improved/cleaned) test set from the CrossWeigh paper [83] called CoNLL++. The current state-of-the-art model on this dataset is from the CrossWeigh paper (also using flair) by Wang et al. [84] has an F1-score of 94.3%. For the model evaluation, we use a model without using pooled-embeddings, CrossWeigh, and training to a max 50 instead of 150 epochs to get a micro F1-score of 93.5%, within 0.7 of a percentage point of the SOTA.

TABLE 3.6: Evaluation Results on CoNLL++ Dataset

	Precision	Recall	F1 Score
LOC	0.955	0.954	0.9547
MISC	0.8397	0.8479	0.8438
ORG	0.9119	0.9172	0.9145
PER	0.9826	0.9759	0.9792
Micro-Average			0.9354
Macro-Average			0.9231

3.2.2 BC5CDR Dataset

3.2.2.1 Spark NLP

The deep neural network design for the NER model in Spark NLP is the BiLSTM-CNN-Char structure, a slightly revised version of the design introduced by Chiu et al. Chiu and Nichols [2016] [13]. It is a NN design that automatically recognizes the word and character-level features employing a hybrid bidirectional LSTM and CNN architecture, dropping the need for most feature engineering steps. In the original framework, the CNN obtains a fixed-length feature vector from character-level features. These vectors are concatenated and supplied to the BLSTM network and later to the output layers for every word. They used a stacked bi-directional RNN with LSTM units to convert word features into named entity tag points. The obtained features of each word are supplied into a forward LSTM network and a backward LSTM network. The output of each network at every time step is decoded by a linear layer and a log-softmax layer into log probabilities for every tag class. These two vectors are summed collectively to produce the final output Chiu and Nichols [2016] [13]. The complete design of the proposed framework in the original paper is represented in Figure 1. In total, 50-dimensional pre-trained word embeddings are applied for word features, 25-dimension character embeddings are applied for char features, and capitalization features are used for case features. They also used lexicons as external information, as suggested in Ratinov and Roth [2009].

In Spark NLP, the authors [42] have adjusted this structureas follows: Habibi et al., in 2017 [34], analyzed the performance of the LSTMCRF strategy on 33 data sets including five distinct entity types with that of best-of-class NER tools and an entity-agnostic CRF implementation. On average, the F1- score of LSTM-CRF is 5% higher than that of the baselines, using WikiPubMed-PMC word embeddings. Adopting a similar NN design, teh authors trained our biomedical word embeddings with the skip-gram model on PubMed abstracts and case studies, as explained in [62], for learning distributed representations of words utilizing contextual information. The

trained word embeddings have 200-dimensions and a vocabulary size of 2.2 million. In order to distinguish the effectiveness of these embeddings, the paper also used 300-dimension pre-trained GloVe embeddings with 6 billion tokens, trained on Wikipedia and [69].

Even though better results were published by Ghaddar and Langlais [29] through robust lexical features, after testing with different parameters and components, the authors chose to eliminate lexical features in order to overcome the complexity and relied on pretrained biomedical embeddings, casing features, and char features through CNN. As expressions are expressed through 2 nested sequences (words chars), a CNN is used so that every character is enclosed in a character embedding matrix of dimension 25. Then, a 1D Convolution layer processes the series of embedded char vectors, followed by a MaxPooling operation.

TABLE 3.7: Test Set Evaluation Results on BC5CDR Dataset

Spark NER	Precision	Recall	F1 Score	Support
B-Disease	0.84	0.86	0.85	960
I-Disease	0.82	0.88	0.85	1087
O	0.99	0.99	0.99	22450
Accuracy			0.98	24497
Micro-Average	0.88	0.91	0.9	24497
Weighted Average	0.98	0.98	0.98	24497

3.2.2.2 BioBERT

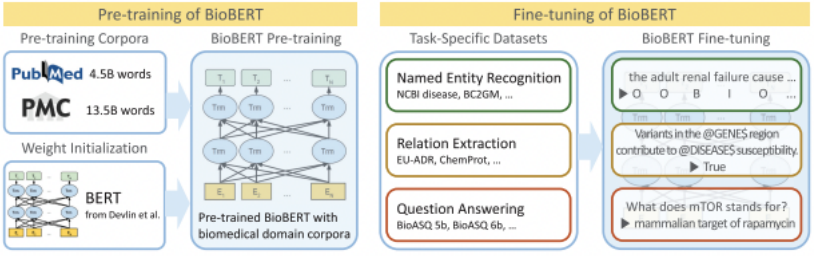

FIGURE 3.2: BioBERT Model.

The Authors of the BioBERT model [45] examine how the pre-trained language model BERT can be accommodated for biomedical corpora. They propose BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining), a domain-specific language design model pre-trained on large-scale biomedical corpora. With about the same design across tasks, BioBERT vastly beats BERT and previous state-of-the-art models in various biomedical text mining tasks when pre-trained on biomedical corpora. While BERT achieves results similar to that of previous state-of-the-art systems, BioBERT significantly beats them on the following

three representative biomedical text mining tasks: biomedical named entity recognition (0.62% F1 score improvement), biomedical relation extraction (2.80% F1 score improvement), and biomedical question answering (12.24% MRR improvement). Their analysis outcomes reveal that pre-training BERT on biomedical corpora aids it in understanding complex biomedical texts.

The model by Eugene Siow [76] follows the model by Nooralahzadeh et al. [66], is trained on top of the BioBERT model as a base and uses Simple Transformers library and has an F1 score of 0.893, which is slightly worse (difference of 0.19) than the current state-of-the-art model in this dataset.

TABLE 3.8: Evaluation results of BioBERT model on BC5CDR dataset

BioBERT BC5CDR	Precision	Recall	F1-Score
	0.877	0.909	0.893

3.2.3 MSRA/SIGHAN2006 Dataset

The Eugene Siow's [76] proposed a model to train/fine-tune a pre-trained chinese BERT model to perform named entity recognition (NER) on the dataset SIGHAN 2006 [46], or commonly known as the MSRA NER dataset. It contains 46,364 samples in the training set and 4,365 samples in the test set. The original workshop/paper for the dataset is by Levow [46]. The current state-of-the-art model on this dataset is the Lattice LSTM from Zhang et al. [93] with an F1-score of 93.2%. The proposed BERT model (with only 1 epoch training) has an F1-score of 93.9%.

TABLE 3.9: Evaluation Results of BERT model on SIGHAN2006 dataset

BERT SIGHAN2006	Precision	Recall	F1-Score
	0.9305	0.9478	0.939

Chapter 4

Reviewing Existing Work

4.1 System Architecture

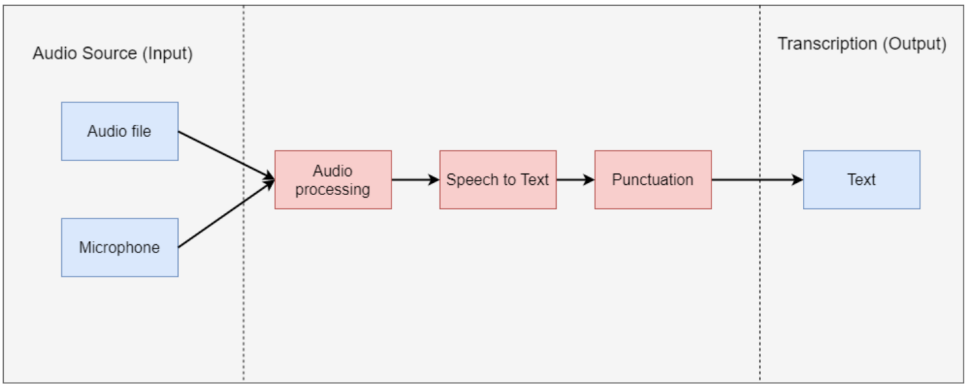

FIGURE 4.1: System Architecture

A system architecture used in this application is the pipe and filter model. The Pipe and Filter is a design pattern that allows for stream or asynchronous processing. This architecture consists of many components, which are referred to as filters and the connector between these filters that called pipes. Filters will process the data and passing them to other filters in the pipeline. The audio source can be either an audio file (.wav) or from a microphone. The audio data is pump into audio processing filter that will divide the audio data into chunks. After which, each chunk is pump into an ASR service to obtain the text and further pump into a punctuation service. Finally, the output will be displayed in the table.

4.2 GUI Components

In this section, it will show the UI of the existing application and the GUI components used in each screen.

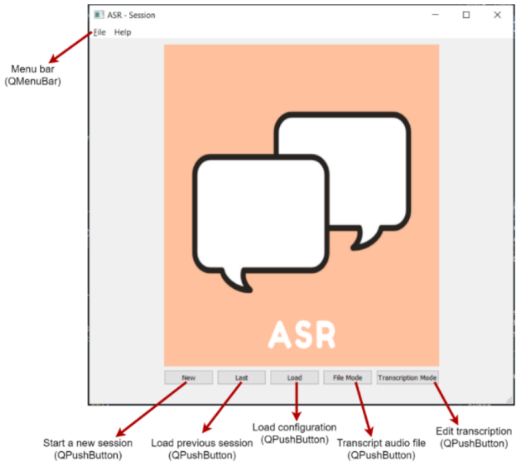

FIGURE 4.2: Intro Screen

FIGURE 4.3: First Screen

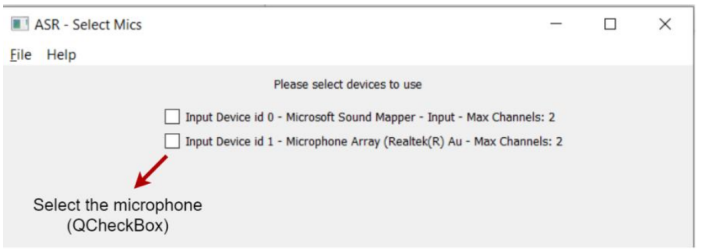

FIGURE 4.4: Second Screen

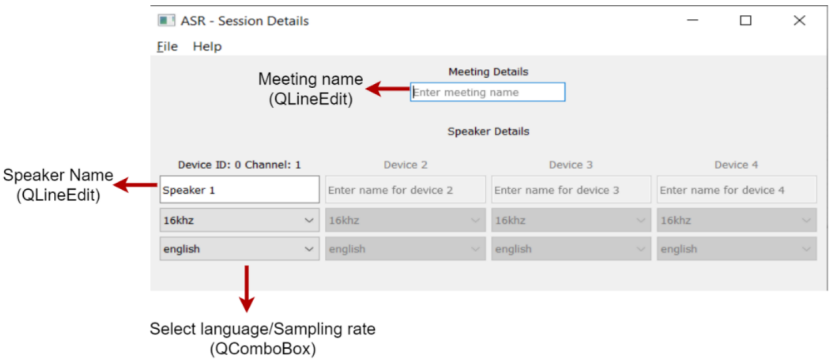

FIGURE 4.5: Third Screen

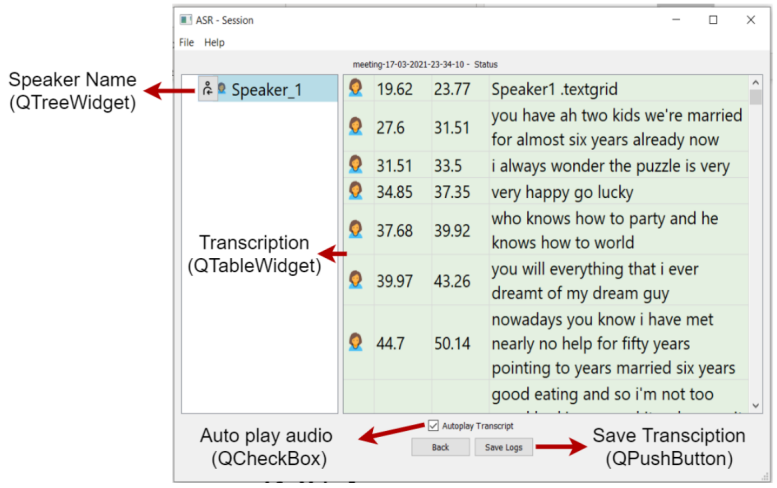

FIGURE 4.6: Fourth Screen

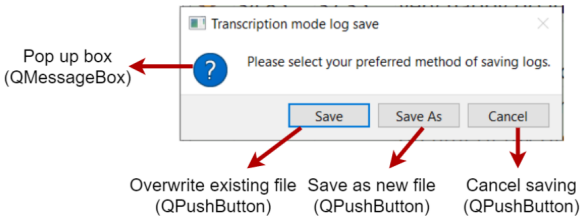

FIGURE 4.7: Saving Transcript

FIGURE 4.8: Audio Playback Control

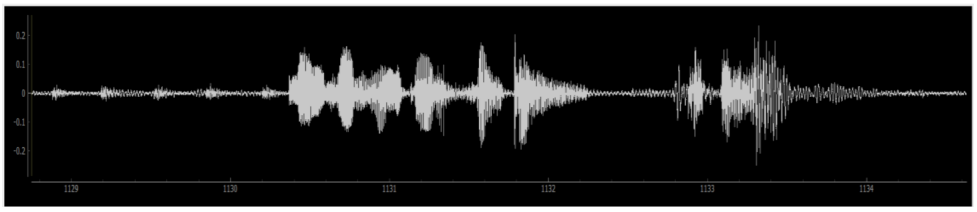

FIGURE 4.9: Audio Waveform Visualization

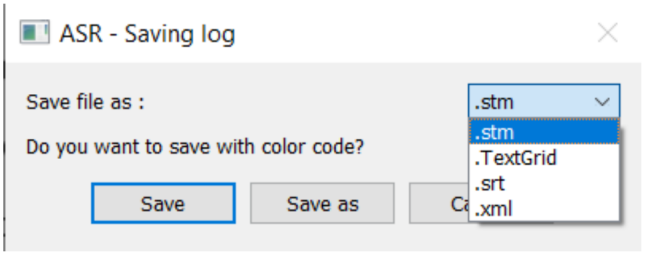

FIGURE 4.10: Support for Cross Format Transcript Saving

4.3 Major Features

4.3.1 Real Rime Speech-to-text

The application can perform real-time transcription that accurately converts speech to text using an ASR service powered by AI Speech Lab. The application will stream the audio data to the ASR service through the computer microphone. Upon receiving results from the ASR service, the application will display the result in a table format that allows the user to edit.

4.3.2 Audio files to Text

The application can also perform speech recognition asynchronously. The user can choose an audio file that he/she wants to transcribe. The application will process the audio files in chunks

and make asynchronous API call to the ASR service. Similarly, the result will be displayed in a table format that allows the user to edit.

4.3.3 Transcription editor

Besides the speech recognition capability, the application has a comprehensive transcript editor. The editor allows the user to choose the transcript and audio files they want to edit. The editor also has support for audio play; as the user is editing the transcript, the respective audio segment will be played. This allows the user to listen to the audio as they edit the transcript

Chapter 5

Implementation

5.1 Software Implementations

5.1.1 Split Audio

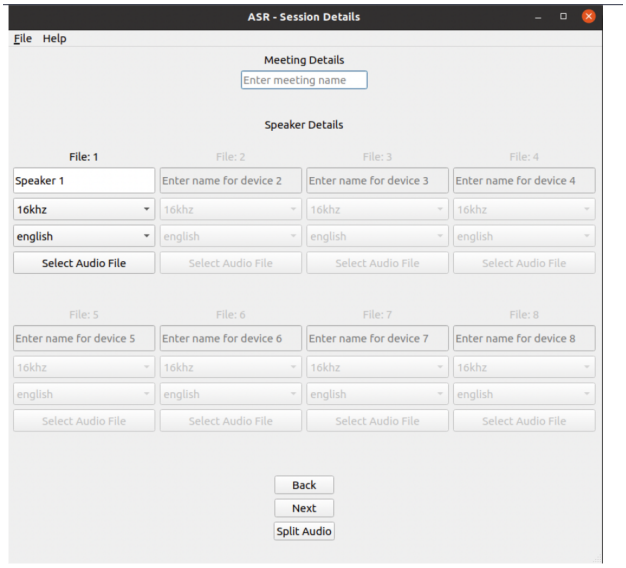

FIGURE 5.1: Split Audio

The current problem with the system is that the system does not support a multi channel file for the file-mode of operation. So, to solve the problem, we have introduced a Split Audio button that splits that channels of a multi-channel audio file into multiple mono-channel files. Split Audio Button Opens a File Selector Window. We need to select the Audio with multiple channels to split into individual audio files.Then, the SOX module splits the audio file into

multiple audio files, one file for a channel If the audio file is file.wav, the split audio files are saved as file-channel-1.wav, file-channel-2.wav ... etc. The audio files are stored in the temp folder in the current directory. The implementation of the split audio button allows the user to split the multi-channel audio file in the application so that different channels of the audio file can be used as audio files for different speakers when operating in the file mode.

5.1.2 Module Analyser

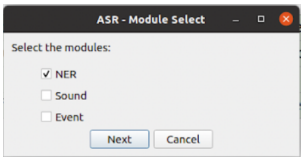

FIGURE 5.2: Module Analyser Select Popup

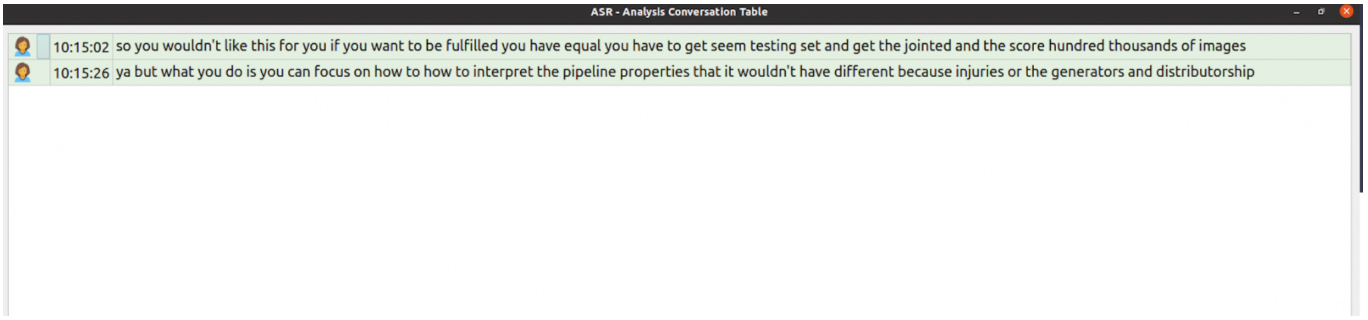

FIGURE 5.3: Module Analyser Table

The implementation of the module analyzer allows the user to display the text of specific rows so that the user can just display the rows corresponding to particular modules that could be implemented in the future (such as event detection).

5.1.3 Named Entity Recognition

In this section, we discuss about the implementation of the named entity recognition for the audio transcription using displaCy Named Entity Recogniser. We use the spacy's inbuilt named entity recogniser tool - displaCy to identify the named entities from the transcript. We then implement a table using PyQT5 table for Named Entity Recognition that tabulates the details of the tokens and its named entity of the transcript that is currently displayed in the conversation table. The implementation of named-entity recognition and entity color coding also allows the user to quickly pinpoint the keyword or phrase and its named entity that they are looking for.

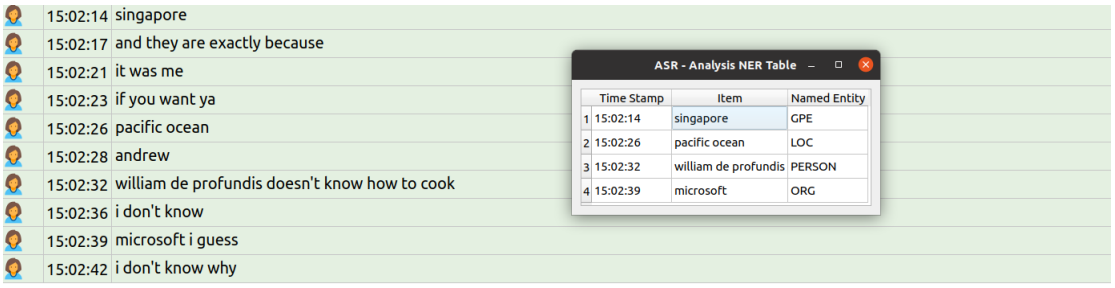

FIGURE 5.4: Named Entity Recognition

5.1.4 Microphone Volume Visualization

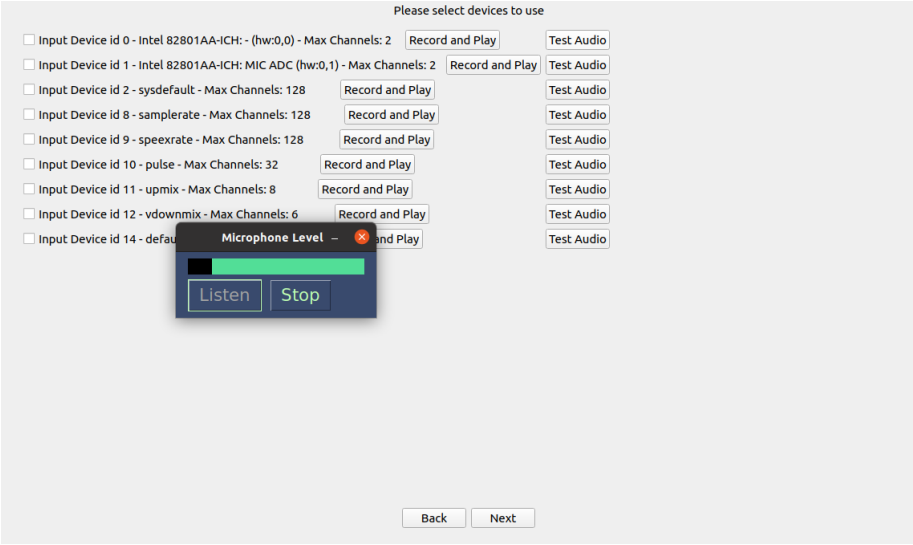

FIGURE 5.5: Microphone Quality Analyser

In this module, we discuss about the implementation of a Microphone Audio Level Visualization while selecting a the microphones for a speaker. The problem with the previous version of the application is that there was no way to check if we are using the right microphone for the audio transcription service. Since the names of the microphones are extracted using pyAudio, there are times where it becomes challenging to identify the exact microphone that we wish to use. So, using a microphone audio quality visualizer and the record-playback feature, the user can test different microphones, comprehend the quality and use the exact microphone for each of the speaker. The implementation is based on the information that we get from the microphone using pyAudio library. Implementing the audio waveform and the record and playback feature for the microphone allows the user to understand the quality of the microphone's audio and select that one that is oh his interest

5.2 Application Components and Flow Charts

In this section, we describe the flow charts corresponding to the working of some of the components of the applications and some flow charts that provide a template to implement specific module in future developments of the product.

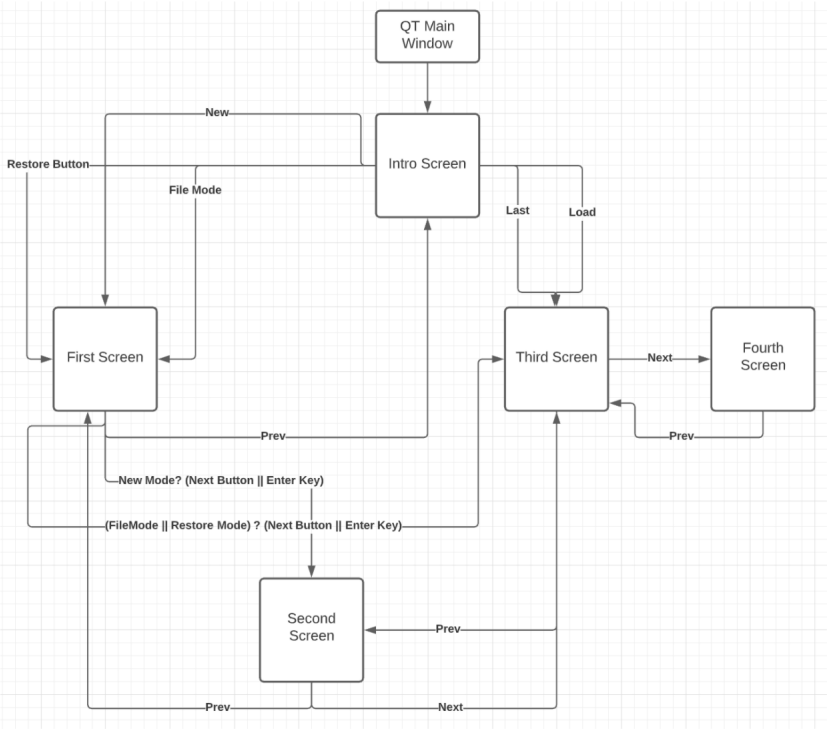

FIGURE 5.6: Application Overview

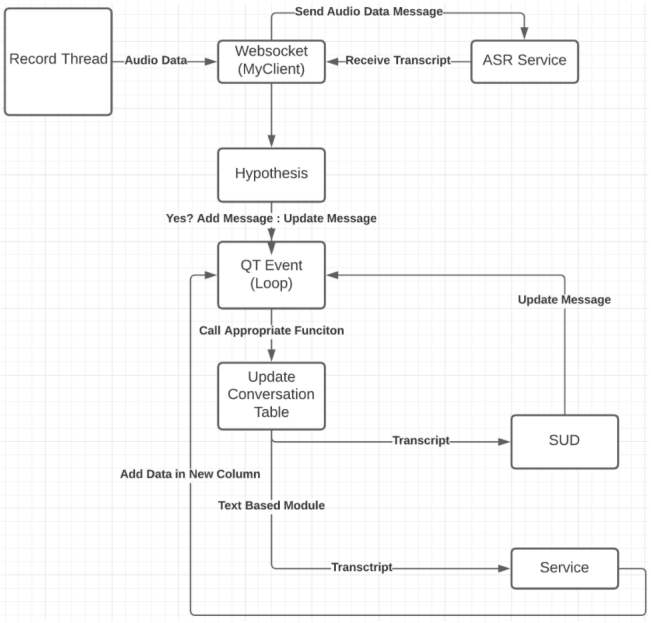

FIGURE 5.7: Text Based Speech Module Technique

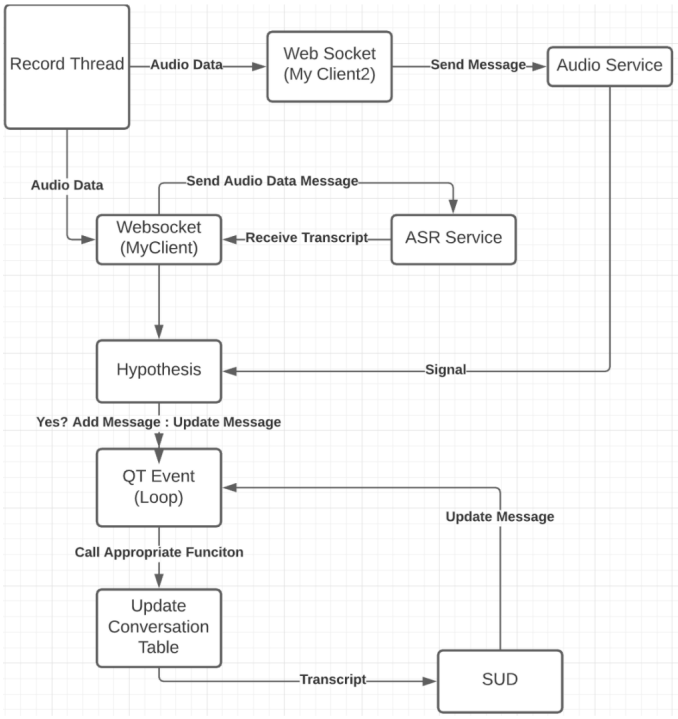

FIGURE 5.8: Audio Based Speech Module

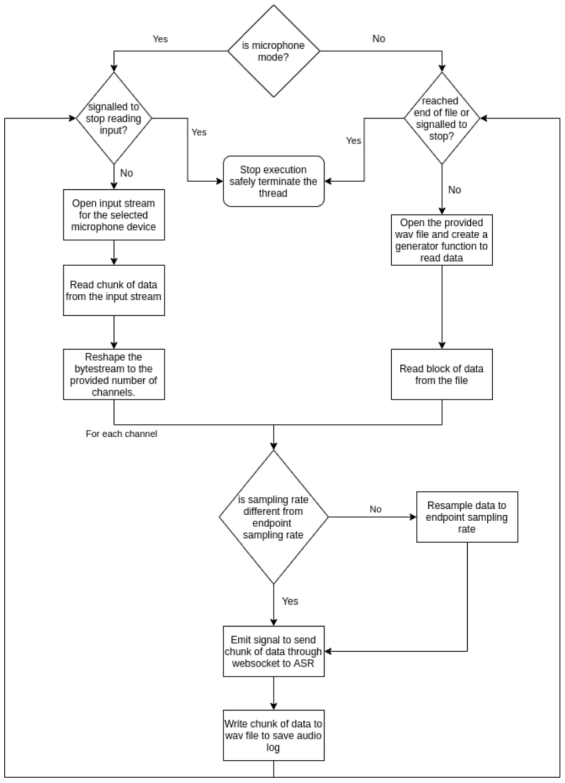

FIGURE 5.9: Working of record thread

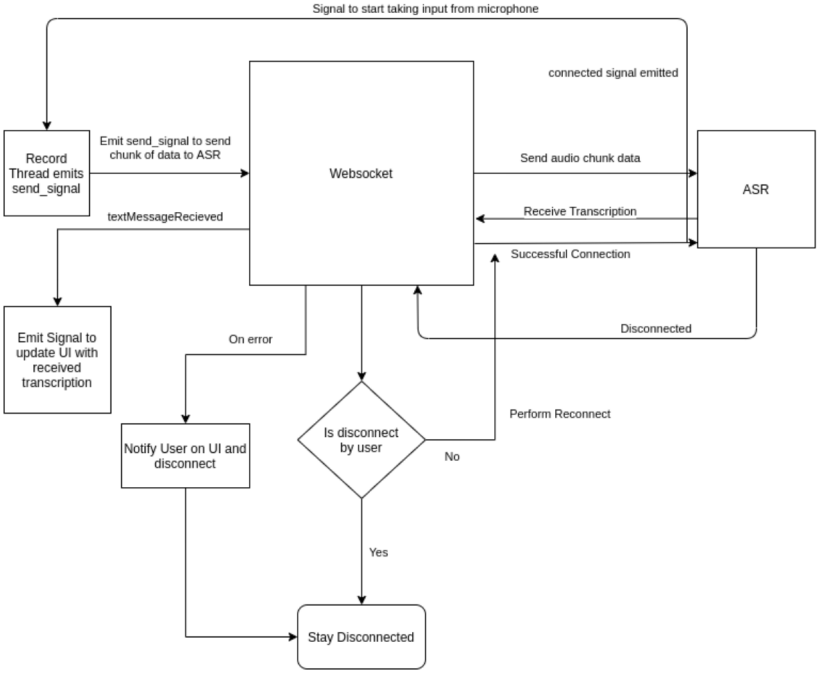

FIGURE 5.10: Working of Speaker web-socket

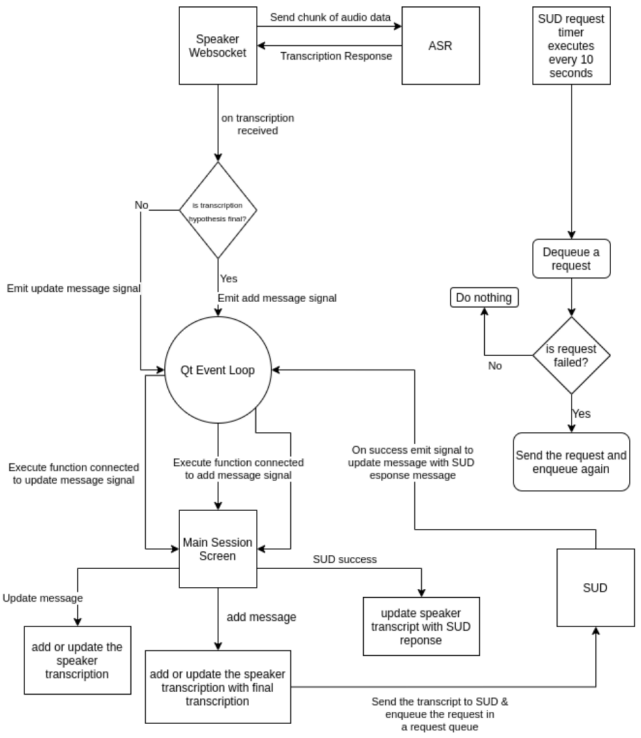

FIGURE 5.11: Update GUI with transcriptions

Chapter 6

Conclusion and Further Work

6.1 Conclusion

In conclusion, the new implementation made to the existing application had improved the efficiency of editing transcripts and the quality of the transcript. The implementation of the split audio button allows the user to split the multi-channel audio file in the application so that different channels of the audio file can be used as audio files for different speakers when operating in the file mode. Implementing the audio waveform and the record and playback feature for the microphone allows the user to understand the quality of the microphone's audio and select that one that is oh his interest. The implementation of the module analyzer allows the user to display the text of specific rows so that the user can just display the rows corresponding to particular modules that could be implemented in the future (such as event detection). The implementation of named-entity recognition and entity color coding also allows the user to quickly pinpoint the keyword or phrase and its named entity that they are looking for.

The use of an agile approach has helped to improve development time and focus on more important features. The weekly sprints allow for constant user feedback and result in a better-quality product. Transcribing software is widely used by users worldwide due to the language barrier. With this application, the user can transcribe in real-time, replacing minute-taking tasks or court-case transcription to transcribe judges and lawyers or even call-center transcription. This application will provide a complete ASR solution that allows the user to transcribe, save, listen and edit.

6.2 Future Work

Due to the application's large scale, some areas are not fully examined, and there are limitations in the current application. In this report, future work should explore the following directions: Improving packaging and deployment Combining short sentences Video-file support Multi-channel audio file support

Bibliography

[1] Mikheev A, Moens M, and Grover C. "Named entity recognition without gazetteers". In: *EACL* (1999), 1–8.

[2] Quimbaya P A et al. "Named entity recognition over electronic health records through a combined dictionary-based approach". In: *Procedia Comput. Sci* 100 (2016), 55–61.

[3] Radford A et al. "Im- proving language understanding by generative pre-training". In: *OpenAI* (2018).

[4] Vaswani A et al. "Attention is all you need". In: *NIPS* (2017), 5998–6008.

[5] Enrique Alfonseca and Suresh Manandhar. "An unsupervised method for general named entity recognition and automated concept discovery". In: *Proceedings of the 1st international conference on general WordNet, Mysore, India.* (2002).

[6] Nasser Alshammari and Saad Alanazi. "The impact of using different annotation schemes on named entity recognition". In: *Egyptian Informatics Journal* 22.3 (2021), pp. 295–302. ISSN: 1110-8665. DOI: https://doi.org/10.1016/j.eij.2020.10.004. URL: https://www.sciencedirect.com/science/article/pii/S1110866520301596.

[7] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), 157–166. DOI: 10.1109/72.279181.

[8] Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. "An algorithm that learns what's in a name." In: *Machine Learning* 34.1/3 (1999), 211–231. DOI: 10.1023/a:1007558221122.

[9] Borthwick and Andrew Eliot. "A maximum entropy approach to named entity recognition." In: (1999).

[10] Aone C et al. "Sra: Description of the ie2 system used for muc-7," in: *MUC-7* (1998).

[11] Nancy Chinchor. "MUC-3 evaluation metrics". In: *Proceedings of the 3rd conference on Message understanding - MUC3 91* (1991). DOI: 10.3115/1071958.1071961.

[12] Jason P.c. Chiu and Eric Nichols. "Named Entity Recognition with Bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4 (2016), 357–370. DOI: 10.1162/tacl_a_00104.

[13] Jason PC Chiu and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 357–370.

[14] Kyunghyun Cho et al. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (2014). DOI: 10.3115/v1/w14-4012.

[15] Sam Coates-Stephens. "The Analysis and Acquisition of Proper Names for the Understanding of Free Text". In: *Computers and the Humanities* 26.5-6 (1992), 441–456. DOI: 10.1007/bf00136985.

[16] Michael Collins and Yoram Singer. "Unsupervised models for named entity classification". In: *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. 1999.* (1999).

[17] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), 273–297. DOI: 10.1007/bf00994018.

[18] James R. Curran and Stephen Clark. "Language independent NER using a maximum entropy tagger". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* - (2003). DOI: 10.3115/1119176.1119200.

[19] Appelt E D et al. "Sri international fastus system: Muc-6 test results and analysis," in: *MUC-6* (1995), 237–248.

[20] Hanisch D et al. "Prominer: rule-based protein and gene entity recognition". In: *BMC Bioinform.* 14th ser. 6.1 (2005).

[21] Nadeau D and Sekine S. "A survey of named entity recognition and classification". In: *Lingvist. Investig* 30.1 (2007), 3–26.

[22] Agirre E et al. "Enriching very large ontologies using the www." In: *Proceedings of the Ontology Learning Workshop, ECAI* (2000).

[23] Jeffrey L. Elman. "Finding Structure in Time". In: *Cognitive Science* 14.2 (1990), 179–211. DOI: 10.1207/s15516709cog1402_1.

[24] Oren Etzioni et al. "Unsupervised named-entity extraction from the Web: An experimental study". In: *Artificial Intelligence* 165.1 (2005), 91–134. DOI: 10.1016/j.artint.2005.03.001.

[25] Christiane Fellbaum. "WordNet". In: *Theory and Applications of Ontology: Computer Applications* (2010), 231–243. DOI: 10.1007/978-90-481-8847-5_10.

[26] Doddington R G et al. "The automatic content extrac- tion (ace) program-tasks, data, and evaluation." In: *LREC* 2 (2004), p. 1.

[27] Krupka G and IsoQuest K. "Description of the NetOwl extractor system as used for muc-7," in: *MUC-7* (2005), 21–28.

[28] Lample G et al. "Neural architectures for named entity recognition." In: *Proceedings NAACL-HLT* (2016).

[29] Abbas Ghaddar and Philippe Langlais. "Robust lexical features for improved neural network named-entity recognition". In: *arXiv preprint arXiv:1806.03489* (2018).

[30] Alex Graves and Jürgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures". In: *Neural Networks* 18.5-6 (2005), 602–610. DOI: 10.1016/j.neunet.2005.06.042.

[31] Ralph Grishman. "The NYU System for MUC-6 or Wheres the Syntax?" In: (1995). DOI: 10.21236/ada460232.

[32] Qipeng Guo et al. "Star-Transformer". In: *Proceedings of the 2019 Conference of the North* (2019). DOI: 10.18653/v1/n19-1133.

[33] Yan H et al. "Tener: Adapting trans- former encoder for name entity recognition". In: (2019). URL: arXiv:1911.04474,.

[34] Maryam Habibi et al. "Deep learning with word embeddings improves biomedical named entity recognition". In: *Bioinformatics* 33.14 (2017), pp. i37–i48.

[35] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[36] Devlin J et al. "Bert:Pre-training of deep bidirectional transformers for language understanding". In: *NAACL-HLT* (2019), 4171–4186.

[37] Yufan Jiang et al. "Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019). DOI: 10.18653/v1/d19-1367.

[38] Zhanming Jie and Wei Lu. "Dependency-Guided LSTM-CRF for Named Entity Recognition". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019). DOI: 10.18653/v1/d19-1399.

[39] Humphreys K et al. In: *University of Sheffield: Description of the lasie-ii system as used for muc-7,* (1998).

[40] Kim, Ji-Hwan, and Philip C Woodland. "A rule-based named entity recognition system for speech input." In: *Sixth International Conference on Spoken Language Processing.* (2000).

[41] Kim, Ji-Hwan, and Philip C. Woodland. "A rule-based named entity recognition system for speech input." In: *Sixth International Conference on Spoken Language Processing.* 2000.

[42] Veysel Kocaman and David Talby. *Biomedical Named Entity Recognition at Scale.* 2020. arXiv: 2011.06315 [cs.CL].

[43] Michal Konkol and Miloslav Konopík. "Segment Representations in Named Entity Recognition". In: *Text, Speech, and Dialogue Lecture Notes in Computer Science* (2015), 61–70. DOI: 10.1007/978-3-319-24033-6_7.

[44] Andrew McCallum Lafferty John and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." In: (2001).

[45] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[46] Gina-Anne Levow. "The third international Chinese language processing bakeoff: Word segmentation and named entity recognition". In: *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing.* 2006, pp. 108–117.

[47] Jiao Li et al. "BioCreative V CDR task corpus: a resource for chemical disease relation extraction". In: *Database* 2016 (2016).

[48] Jing Li et al. "A Survey on Deep Learning for Named Entity Recognition". In: *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1. DOI: 10.1109/tkde.2020.2981314.

[49] Xiaoya Li et al. "A Unified MRC Framework for Named Entity Recognition". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020). DOI: 10.18653/v1/2020.acl-main.519.

[50] Xiaoya Li et al. "Dice Loss for Data-imbalanced NLP Tasks". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020). DOI: 10.18653/v1/2020.acl-main.45.

[51] Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. "SVM Based Learning System for Information Extraction". In: *Lecture Notes in Computer Science Deterministic and Statistical Methods in Machine Learning* (2005), 319–339. DOI: 10.1007/11559887_19.

[52] Shengyu Liu et al. "Effects of Semantic Features on Machine Learning-Based Drug Name Recognition Systems: Word Embeddings vs. Manually Constructed Dictionaries". In: *Information* 6.4 (2015), 848–865. DOI: 10.3390/info6040848.

[53] Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. "Towards Improving Neural Named Entity Recognition with Gazetteers". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019). DOI: 10.18653/v1/p19-1524.

[54] Yijin Liu et al. "GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019). DOI: 10.18653/v1/p19-1233.

[55] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[56] Ying Luo, Fengshun Xiao, and Hai Zhao. "Hierarchical Contextualized Representation for Named Entity Recognition". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05 (2020), 8441–8448. DOI: 10.1609/aaai.v34i05.6363.

[57] Xuezhe Ma and Eduard Hovy. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2016). DOI: 10.18653/v1/p16-1101.

[58] Robert Malouf. "A comparison of algorithms for maximum entropy parameter estimation". In: *proceeding of the 6th conference on Natural language learning - COLING-02* (2002). DOI: 10.3115/1118853.1118871.

[59] Mónica Marrero et al. "Named Entity Recognition: Fallacies, challenges and opportunities". In: *Computer Standards Interfaces* 35.5 (2013), 482–489. DOI: 10.1016/j.csi.2012.09.004.

[60] Andrew Mccallum and Wei Li. "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -* (2003). DOI: 10.3115/1119176.1119206.

[61] Paul Mcnamee and James Mayfield. "Comparing cross-language query expansion techniques by degrading translation resources". In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR 02* (2002). DOI: 10.1145/564376.564406.

[62] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[63] Valerie Mozharova and Natalia Loukachevitch. "Two-stage approach in Russian named entity recognition". In: *2016 International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT)*. IEEE. 2016, pp. 1–6.

[64] Kitaev N and Klein D. "Constituency parsing with a self- attentive encoder". In: *ACL* (2018), 2675–2685.

[65] David Nadeau, Peter D. Turney, and Stan Matwin. "Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity". In: *Advances in Artificial Intelligence Lecture Notes in Computer Science* (2006), 266–277. DOI: 10.1007/11766247_23.

[66] Farhad Nooralahzadeh, Jan Tore Lønning, and Lilja Øvrelid. "Reinforcement-based denoising of distantly supervised NER with partial annotation". In: *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 225–233. DOI: 10.18653/v1/D19-6125. URL: https://aclanthology.org/D19-6125.

[67] Liu J P et al. "Generating wikipedia by summarizing long sequences". In: (2018). URL: arXivpreprintarXiv:1801.10198.

[68] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014). DOI: 10.3115/v1/d14-1162.

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

[70] Collobert R et al. "Naturallanguage processing (almost) from scratch." In: *Proceedings NAACL-HLT* (2016).

[71] Pascanu R, Mikolov T, and Bengio Y. "On the difficulty of training recurrent neural networks." In: *International Conference on Machine Learning* (2013), 1310–1318.

[72] L.f. Rau. "Extracting company names from text". In: *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application* (). DOI: 10.1109/caia.1991.120841.

[73] Sekine S and Nobata C. "Definition, dictionaries and tagger for extended named entity hierarchy." In: *Proceedings of the language resources and evaluation conference (LREC)* (2004), 1997–1980.

[74] Erik F. Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 shared task". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* - (2003). DOI: 10.3115/1119176.1119195.

[75] Yusuke Shinyama and Satoshi Sekine. "Named entity discovery using comparable news articles". In: *Proceedings of the 20th international conference on Computational Linguistics - COLING 04* (2004). DOI: 10.3115/1220355.1220477.

[76] Eugene Siow. *Practical Machine Learning: A Collection of Machine Learning Experiments in Notebooks*. Available at: https://github.com/eugenesiow/practical-ml. 2020. URL: https://github.com/eugenesiow/practical-ml.

[77] Michael Weidlich Tim Rockta schel Torsten Huber and Ulf Leser. "The impact of domain-specific features on the performance of identifying and classifying mentions of drugs." In: (2013), 356–363.

[78] Asahi Ushio and Jose Camacho-Collados. "T-NER: An All-Round Python Library for Transformer-based Named Entity Recognition". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. 2021, pp. 53–62.

[79] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[80] Black J W, Rinaldi F, and Mowatt D. "Facile: Description of the ner system used for muc-7". In: *MUC-7* (1998).

[81] Ling W et al. "Not all contexts are created equal: Better word representations with variable attention." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* (2015), 1367–1372.

[82] Waltl et al. "Rule-based information extraction: Advantages, limitations, and perspectives". In: *Jusletter IT* 2 (2018).

[83] Zihan Wang et al. "Crossweigh: Training named entity tagger from imperfect annotations". In: *arXiv preprint arXiv:1909.01441* (2019).

[84] Zihan Wang et al. "CrossWeigh: Training Named Entity Tagger from Imperfect Annotations". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5154–5163. DOI: 10.18653/v1/D19-1519. URL: https://aclanthology.org/D19-1519.

[85] Ralph Weischedel et al. "Ontonotes release 5.0 ldc2013t19". In: *Linguistic Data Consortium, Philadelphia, PA* 23 (2013).

[86] Congying Xia et al. "Multi-grained Named Entity Recognition". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019). DOI: 10.18653/v1/p19-1138.

[87] LeCun Y et al. "Backpropagation applied to handwritten zip code recognition." In: *Neural Comput* 1.4 (1989), 541–551.

[88] Vikas Yadav and Steven Bethard Bethard. "A survey on recent advances in named entity recognition from deep learning models." In: (2019). URL: arXivpreprintarXiv:1910.11470.

[89] Ikuya Yamada et al. "Luke: deep contextualized entity representations with entity-aware self-attention". In: *arXiv preprint arXiv:2010.01057* (2020).

[90] Huang Z, Xu W, and Yu K. "Bidirectional lstm-crf models for sequence tagging." In: (2015). DOI: arXiv:1508.01991.

[91] Yang Z, Salakhutdinov R, and Cohen W. "Multi-task cross-lingual sequence tagging from scratch." In: (2016). URL: http://arxiv.org/abs/1603.06270.

[92] Shaodian Zhang and Noémie Elhadad. "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts". In: *Journal of Biomedical Informatics* 46.6 (2013), 1088–1098. DOI: 10.1016/j.jbi.2013.08.004.

[93] Yue Zhang and Jie Yang. "Chinese NER using lattice LSTM". In: *arXiv preprint arXiv:1805.02023* (2018).

[94] Guodong Zhou and Jian Su. "Named entity recognition using an HMM-based chunk tagger". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL 02* (2001). DOI: 10.3115/1073083.1073163.

[95] Jingwei Zhuo et al. "Segment-Level Sequence Modeling using Gated Recursive Semi-Markov Conditional Random Fields". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2016). DOI: 10.18653/v1/p16-1134.