Domain Adaptation of Named Entity Recognition for Plant Health Monitoring

Mariya Borovikova

université PARIS-SACLAY

Domain Adaptation of Named Entity Recognition for Plant Health Monitoring

Adaptation de la reconnaissance d'entités nommées au domaine de la santé des plantes

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate School : Informatique et sciences du numérique.
Référent : Faculté des sciences d'Orsay

Thèse préparée dans les unités de recherche **MaIAGE (Université Paris-Saclay, INRAE)** et **TETIS (AgroParisTech, CNRS, CIRAD, INRAE)**,
sous la direction de **Claire NEDELLEC**, Directrice de recherche INRAE,
la co-direction de **Mathieu ROCHE**, Directeur de recherche CIRAD
et le co-encadrement de **Arnaud FERRE**, Chargé de recherche INRAE
et **Robert BOSSY**, Ingénieur de recherche INRAE

Thèse soutenue à Paris-Saclay, le 13 décembre 2024, par

Mariya BOROVIKOVA

Composition du jury

Membres du jury avec voix délibérative

Aurélie NÉVÉOL Directrice de Recherche, CNRS	Président
Gaël DIAS Professeur, Université de Caen Basse-Normandie	Rapporteur & Examinateur
Laure SOULIER Maîtresse de conférence, Sorbonne Université	Rapportrice & Examinatrice
Thierry CHARNOIS Professeur, LIPN, Université Sorbonne Paris Nord	Examinateur
Adrien COULET Chargé de recherche, Inria Paris	Examinateur

THESE DE DOCTORAT
NNT : 2024UPASG105

université
PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies
de l'information et de
la communication (STIC)

Titre: Adaptation de la reconnaissance d'entités nommées au domaine de la santé des plantes
Mots clés: Reconnaissance des Entités Nommées, Santé des Plantes, Apprentissage automatique, Adaptation au Domaine, Modèles de Langue, Zero-Shot Learning, Traitement Automatique des Langues

Résumé:
La complexité croissante des écosystèmes agricoles et le besoin urgent de surveillance efficace de la santé des plantes rendent nécessaires des solutions technologiques avancées pour traiter les données textuelles. Située dans le cadre du projet BEYOND, cette thèse répond à ces besoins en améliorant les systèmes de reconnaissance d'entités nommées (REN) adaptés au domaine de la santé des plantes. Reconnaissant les limites des approches traditionnelles, cette recherche intègre des stratégies d'adaptation au domaine.

La principale contribution de cette thèse réside dans le développement et l'affinement de méthodes destinées à améliorer l'adaptabilité des systèmes REN dans la reconnaissance d'informations liées à la santé des plantes, telles que les maladies végétales, les organismes nuisibles, les plantes et les lieux. En exploitant des techniques avancées d'apprentissage automatique, la thèse montre comment les systèmes REN peuvent être appliqués à la surveillance de la santé des plantes sans nécessiter d'adaptation explicite.

Pour relever ces défis, la thèse a trois objectifs. Le premier objectif vise à adapter un modèle de langue au domaine de la santé des plantes, en se concentrant sur son futur application à la tâche de REN. Les questions de recherche pour cet objectif explorent les méthodes par lesquelles un modèle de langage peut être ajusté pour prendre en considération la terminologie spécialisée et les nuances contextuelles de la santé des plantes.

Le deuxième objectif vise à développer une méthode permettant à un système de REN existant de reconnaître de nouvelles entités nommées sans qu'il soit nécessaire de procéder à un ajustement explicite. Les questions posées ici cherchent à découvrir comment des systèmes existants peuvent reconnaître et catégoriser de manière autonome de nouvelles entités sur la base de modèles et de caractéristiques linguistiques trouvés dans des données textuelles non structurées, réduisant ainsi la dépendance de vastes ensembles de données annotées.

Le troisième objectif se concentre sur la conception d'un système d'adaptation complet spécifiquement adapté au domaine de la santé des plantes, et s'appuie sur les résultats des deux premiers objectifs. La question de recherche pour cet objectif est de savoir si la recombinaison de différents modules du même système, chacun ajusté séparément pour des tâches similaires, permet d'obtenir des résultats comparables à ceux d'un modèle réglé de manière classique. Ce processus implique de combiner le modèle de langue avec les stratégies de reconnaissance d'entités nommées dans un système robuste et évolutif capable de gérer les scénarios de données complexes typiques de la surveillance de la santé des plantes.

Sur le plan méthodologique, la thèse adopte une approche double. D'une part, elle ajuste les modèles de langue grâce au masquage de mots-clés, focalisant le processus d'apprentissage sur le vocabulaire spécifique au domaine pour capturer les particularités linguistiques de la santé des plantes. D'autre part, elle améliore la reconnaissance des entités nommées grâce à l'intégration de représentations sémantiques obtenues à partir de descriptions textuelles des types d'entités. Cette méthode permet à l'algorithme de reconnaître des types d'entités non rencontrés durant l'apprentissage et de s'adapter facilement à de nouvelles applications. Cette méthodologie est ensuite appliquée aux données sur la santé des plantes, combinant les deux approches.

Cette recherche contribue à l'avancement théorique dans le domaine de la REN et présente des implications pratiques, fournissant des outils susceptibles de conduire à une prise de décision plus informée face aux menaces phytosanitaires. Les orientations futures de ce travail incluent l'affinement des approches basées sur les lexiques, l'intégration de données multimodales et l'amélioration des définitions d'entités pour perfectionner davantage la précision et l'applicabilité des systèmes REN dans des domaines spécialisés tels que la santé des plantes.

Title: Domain Adaptation of Named Entity Recognition for Plant Health Monitoring

Keywords: Named Entity Recognition, Plant Health, Machine Learning, Domain Adaptation, Language Models, Zero-Shot Learning, Natural Language Processing

Abstract: The increasing complexity of agricultural ecosystems and the urgent need for effective plant health monitoring necessitate advanced technological solutions for processing textual data. Situated within the BEYOND project, this thesis addresses these needs by advancing Named Entity Recognition (NER) systems tailored to the plant health domain. Considering the limitations of traditional NER approaches, this research innovates by integrating domain-specific adaptation strategies.

The core contribution of this thesis is the development and refinement of methods to enhance the adaptability of NER systems in recognizing information related to plant health, such as diseases, pests, plants, and locations. By leveraging advanced machine learning techniques, the thesis demonstrates how NER systems can be applied to plant health monitoring without explicit adaptation.

Methodologically, the thesis employs a dual approach. Firstly, it refines language models through Keyword Masking, focusing the training process on domain-relevant vocabulary to capture the specific linguistic features of the plant health domain. Secondly, it enhances entity recognition via semantic entity representations derived from textual descriptions of entity types. This approach enables the algorithm to identify entity types not seen during training, facilitating seamless adaptation to new applications. Finally, this methodology is applied to Plant Health data, combining both approaches for robust analysis.

This research contributes theoretical advancements to the field of NER and offers practical implications for agricultural practices. It provides tools that can lead to more informed decision-making responses to plant health threats. Future directions for this work include refining lexicon-based approaches, integrating multimodal data, and enhancing the entity types definitions to further improve the precision and applicability of NER systems in specialized domains such as plant health.

Acknowledgments

This journey has been one of the most challenging yet rewarding experiences of my life, and it would not have been possible without the support, guidance, and encouragement of many individuals and organizations.

First and foremost, I would like to express my deepest gratitude to my supervisors, Claire Nédellec, Mathieu Roche, Arnaud Ferré, and Robert Bossy, for their unwavering support, insightful guidance, and immense patience throughout this journey. Your expertise and mentorship have been invaluable in shaping my research and my growth as a researcher.

I am also deeply grateful to my thesis committee members, Antoine Doucet and Blaise Hanczar, for their time, constructive feedback, and thought-provoking questions, which significantly enhanced the quality of this work.

Special thanks to my colleagues at MaIAGE and TETIS for their camaraderie, insightful discussions, and countless moments of laughter that made this journey enjoyable. I am particularly grateful to Arnaud Ferré, Christian Poirier, Cyprien Guérin, and Solène Pety for their technical assistance and encouragement during challenging times.

I am thankful to ANR BEYOND for providing the financial support that made this research possible. Additionally, I extend my appreciation to Juliette Degrouard and Annie Huguet for their help and efficiency in navigating the logistical aspects of this project.

A special word of thanks to the Saclay-IA platform of Université Paris-Saclay for providing GPU resources through its Lab-IA cluster.

To my family and friends, who have been my anchor throughout this journey, thank you for your unconditional support, encouragement, and understanding. Your belief in me has been a constant source of strength.

Finally, to all the unsung heroes who have contributed to this thesis in ways big and small, whether through a kind word, a helpful suggestion, or a shared coffee break, I am deeply grateful.

This thesis is a testament to the incredible support network I have been fortunate to have around me.

Contents

1 Introduction **15**

1.1 Research Context and Project Contribution . . . 15
1.2 Motivation and Challenges in Named Entity Recognition for Plant Health . . . 16
1.3 Research Objectives and Questions . . . 18
1.4 Methodological Approach . . . 19
1.5 Thesis Structure . . . 20

2 Background and related work **21**

2.1 Fundamentals of Machine Learning for NLP . . . 21
2.1.1 Supervised Learning . . . 22
2.1.2 Unsupervised and semi-supervised methods . . . 35
2.1.3 Few-shot Learning . . . 35
2.1.4 Adaptive Learning Methods . . . 36
2.2 Language Modeling . . . 36
2.2.1 Language Models pre-training and fine-tuning . . . 36
2.2.2 Early Developments . . . 37
2.2.3 Statistical Models . . . 38
2.2.4 Neural Language Models . . . 39
2.2.5 Domain-specific Language Models . . . 44
2.3 Named Entity Recognition . . . 46
2.3.1 Task Definition . . . 46
2.3.2 Evaluation . . . 47
2.3.3 Datasets . . . 48
2.3.4 Annotation Standards . . . 51
2.3.5 Techniques and Models Overview . . . 53
2.4 Named Entity Recognition and Domain Adaptation for Plant Health . . . 65
2.4.1 Challenges Specific to Plant Health NER . . . 66
2.4.2 Data Sources and Annotation for Plant Health NER . . . 67
2.4.3 Technological Approaches and Models . . . 71
2.5 Conclusion . . . 72

3 Language Model Domain Adaptation: A KeyWord Masking method **75**

3.1 Methodology . . . 75
3.1.1 Masking strategy . . . 77
3.1.2 Datasets . . . 77
3.1.3 Domain-specific terms lists . . . 80
3.1.4 Evaluation method . . . 82
3.2 Experiments . . . 83

3.2.1 Baselines . . . 83
3.2.2 Implementation details . . . 83
3.2.3 Results . . . 84
3.3 Discussion . . . 89
3.3.1 General remarks . . . 89
3.3.2 Entity Type-Specific Observations . . . 89
3.3.3 Concerns over Overfitting . . . 91
3.3.4 Errors Analysis . . . 91
3.4 Conclusion . . . 92

4 Zero-Shot Named Entity Recognition **95**
4.1 Methodology . . . 95
4.1.1 General Pipeline . . . 95
4.1.2 Datasets . . . 96
4.1.3 Semantic representation of entity types . . . 98
4.1.4 Model description . . . 100
4.2 Experiments . . . 101
4.2.1 Baselines . . . 101
4.2.2 Experimental setup . . . 101
4.2.3 Evaluation method . . . 102
4.2.4 Results . . . 103
4.3 Discussion . . . 108
4.4 Conclusion . . . 111

5 Named Entity Recognition domain adaptation for plant health **113**
5.1 Methodology . . . 113
5.1.1 General pipeline . . . 113
5.1.2 Datasets . . . 114
5.1.3 Lexicon filter . . . 115
5.2 Experiments . . . 118
5.2.1 Experimental setup . . . 118
5.2.2 Results . . . 119
5.3 Discussion . . . 120
5.3.1 Insights from the EPOP dataset . . . 120
5.3.2 Potential impact on Plant Health Epidemiology Monitoring . . . 124
5.3.3 Future Directions . . . 125
5.4 Conclusion . . . 127

6 Conclusion **129**
6.1 Summary of findings . . . 129
6.2 Perspectives . . . 131

List of Figures

2.1 **Schematic Representation of a Single Neuron in a Neural Network.** This diagram depicts a single neuron where input values $x_1, x_2, \ldots, x_n$ are each multiplied by corresponding weights $w_1, w_2, \ldots, w_n$. The weighted inputs are then summed to produce the output y, demonstrating the fundamental operation within a neural network neuron. 24

2.2 **Feed-Forward Neural Network.** This diagram illustrates a standard neural network consisting of multiple layers. Each layer is depicted with a set number of neurons, represented as circles. The first layer contains n neurons, the second layer m neurons, and the final layer k neurons, showcasing the structure of a typical feed-forward network architecture. 24

2.3 **Comparison of Hidden Layers in FFNNs and RNNs.** This figure illustrates the key differences in network architecture between Feed-Forward Neural Networks (FFNNs) and Recurrent Neural Networks (RNNs). In the left panel, an FFNN module is depicted where connections between layers propagate strictly forward without any feedback loops; x_t represents the input at time step t, and y_t denotes the output or hidden state at the same step. The right panel illustrates an RNN module, highlighting its recurrent connections that allow feedback from the network's previous output states (h_{t-1})sequences. This setup enables the RNN to maintain a continuous flow of information, where x_t is the input, y_t is the output and h_t is the hidden state at time t. 25

2.4 **Comparison of RNN and LSTM Cell Architectures.** This diagram contrasts the internal structure and processing flow within a standard Recurrent Neural Network (RNN) cell and a Long Short-Term Memory (LSTM) cell. It details the simple recurrent loop of the RNN cell alongside the complex gate mechanisms of the LSTM. 26

2.5 **Comparison of LSTM and BiLSTM Architectures.** This diagram illustrates the architectural differences between Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) networks. On the left, the LSTM processes input sequences (x_1, x_2, ..., x_n) in a forward direction, with each LSTM unit outputting a corresponding y value (y_1, y_2, ..., y_n) after passing through an activation function F. On the right, the BiLSTM processes input sequences both forward and backward, allowing the network to incorporate information from both past and future contexts within the sequence. Each unit in the BiLSTM consists of two LSTMs, one for each direction, whose outputs are combined at each time step to produce the final output. 28

2.6 **Sequence to Sequence model architecture.** The lower row represents the input sequence $x_1, x_2, \ldots, x_n$, processed by the encoder. The upper row represents the output sequence $y_1, y_2, ..., xy_m$. The circles in the middle denote the network units that can be network units, such as RNN or LSTM units as described in Sections 2.1.1 and 2.1.1 respectively. 29

2.7 **Overview of the Transformer Model Architecture.** This schema, taken from the original paper [Vaswani et al., 2017], illustrates the main components of the Transformer model, highlighting both the encoder and decoder structures. Each module consists of multiple layers repeated N times. The encoder processes input embeddings, enhanced with positional encoding, through multi-head attention and feed-forward networks. The decoder follows a similar architecture but incorporates an additional masked multi-head attention layer to prevent the influence of future context on the predictions. . 31

2.8 **Multi-Head Attention Mechanism in a Transformer Model.** This diagram depicts the multi-head attention mechanism, where each "head" processes the input data independently. Bold arrows represent linear transformations, applying a unique set of weights to the input. Dashed arrows indicate the extraction of specific rows from the matrix without any transformation. After processing through a softmax layer to calculate attention weights, the outputs from all heads are concatenated, forming a comprehensive final output that enhances the model's ability to focus on various aspects of the input data simultaneously. 33

2.9 **Illustration of a Conditional Random Field Model.** This figure depicts the basic structure of a CRF model, where x_i and y_i represent the input and output variable nodes, respectively, for each position in a sequence from 1 to n. The black squares in the diagram denote the transition feature function f, as outlined in the formulas presented in this section. 34

2.10 **Named Entities Recognition task.** This example illustrates labeled named entities in a textual passage. 46

3.1 **Overview of the KeyWord Masking Strategy.** This schema illustrates the KWM approach to MLM task, where mentions of specific entity types are selectively masked in the text. The Language Model is then fine-tuned to reconstruct these mentions. 76

3.2 **Example of BRAT annotation from our test corpus for named entity recognition in the plant health domain.** Various entity types are annotated, including plants (e.g., "Cavendish Bananas"), pests (e.g., "Fusarium oxysporum"), and locations (e.g., "Asia", "Latin America"). 79

3.3 **Accuracy comparison of Standard and KWM masking strategies for Bacteria Biotope dataset.** 87

3.4 **Accuracy comparison of Standard and KWM masking strategies for Geovirus dataset.** 87

3.5 **Accuracy comparison of Standard and KWM masking strategies for EPOP dataset.** 87

3.6 **Perplexity comparison of Standard and KWM masking strategies for Bacteria Biotope dataset.** 88

3.7 **Perplexity comparison of Standard and KWM masking strategies for Geovirus dataset.** 88

3.8 **Perplexity comparison of Standard and KWM masking strategies for EPOP dataset.** 88

4.1 **Overview of the Semantic Named Entity Recognition Model.** This diagram illustrates how the model encodes entity types, such as "Cuisine", using Sentence-BERT with descriptions sourced from Wikipedia. Concurrently, input text is encoded using BERT. Both sets of embeddings are processed through linear layers, combined via a dot product, and then passed through a classifier to identify specific entities like "Indian", which is a type of cuisine. The colors in the diagram—yellow for Language Models, gold for the dot product, red for entities, and blue for transformation layers—visually differentiate the stages of data processing. 96

4.2 **Semantic Representation Process.** This figure illustrates the process of constructing semantic representations for various entity types used in Named Entity Recognition on the example of "Habitat", "Genre", and "Review" entities. The corresponding Wikipedia descriptions are selected for each entity. For ambiguous entity types like "Genre", where the specific domain (e.g., "Film genre") is necessary, a precise article is chosen to match the context of the dataset. Each description is then processed through Sentence-BERT, which transforms the textual content into latent representations. 100
4.3 **Integration of predictions.** This diagram visually represents the process of consolidating entity predictions for multiple entity types into a unified output. This method enables a holistic evaluation of our model's performance across various entity types. 102

5.1 **Overview of the Integrated NER Methodology for Plant Health.** This figure illustrates the application of a zero-shot Named Entity Recognition strategy (SemNER), as described in Chapter 4, to the Plant Health domain. It employs a domain-specific Language Model that was fine-tuned using the KeyWord Masking strategy outlined in Chapter 3. 114
5.2 **Performance Metrics for "Plant" Entities Using Different Lexicon Length Filters.** 116
5.3 **Performance Metrics for "Pest" Entities Using Different Lexicon Length Filters.** 116
5.4 **Performance Metrics for "Disease" Entities Using Different Lexicon Length Filters.** 117
5.5 **Performance Metrics for "Location" Entities Using Different Lexicon Length Filters.** 117

List of Tables

2.1 **Summary of datasets related to Plant health monitoring and agriculture for NER relevant to our study.** This table includes details on the number of mentions and documents, types of documents, and entities types for each dataset. . . . 67

2.2 **Summary of datasets for geographic NER relevant to our study.** This table includes details on the number of entities and documents, types of documents, and geographical coverage for each dataset. 69

3.1 **Entity Types Across Different Datasets.** This table presents the total and unique counts of various entity types within the EPOP, Bacteria Biotope (development set), and GeoVirus datasets. 81

3.2 **Overview of Domain-Specific lexicons for Masked Language Modeling** 81

3.3 **Accuracy Performance Across Datasets.** This table presents the accuracy metrics for BERT and BioBERT models tested across various datasets for MLM task. It highlights the model performance under no fine-tuning, standard masking, and KeyWord Masking conditions, with the best scores marked in bold for each dataset and model combination. Each row in 'Masked tokens" column indicates which entities were masked during evaluation. Rows labeled "All" indicate that all entity types from the dataset in question were masked, while rows labeled "Random" reflect accuracy when tokens were masked at random, without targeted entity masking. This table provides a comparative view of how each fine-tuning strategy influences model accuracy. 85

3.4 **Perplexity Performance Across Datasets.** This table details the perplexity scores of the BERT and BioBERT models on MLM task under different fine-tuning approaches: without fine-tuning, with standard masking approach, and with KeyWord Masking approach. The metrics are presented across various datasets, highlighting the best performances in bold. Each score represents an average of results from ten training iterations, alongside the corresponding standard deviations. . 86

3.5 **Named Entity Recognition Performance Metrics.** This table compares the precision, recall, and F-measure of BERT models under three different masking approaches: without fine-tuning, standard masking, and KeyWord Masking. Results are shown for different entity types across three datasets: EPOP, Bacteria Biotope (BB), and GeoVirus. The highest scores for each metric across the testing conditions are highlighted in bold, demonstrating the effectiveness of the masking strategies in varying contexts. 90

4.1 **Datasets Overview.** This table presents counts of total and unique entity types for each dataset, detailed for both training and testing partitions. It also includes counts of overlapping unique entities across these partitions. 97
4.2 **Overall comparison of F-score across all the datasets.** . . . 104
4.3 **Performance metrics on MIT Restaurants dataset.** 104
4.4 **Performance metrics on MIT Movies dataset.** 105
4.5 **Performance metrics on CoNLL-2003 dataset.** 106
4.6 **Performance metrics on NCBI Diseases dataset.** 106
4.7 **Performance metrics on Bacteria Biotope dataset.** 107

5.1 **Performance metrics on EPOP dataset** 119

1 - Introduction

1.1 . Research Context and Project Contribution

The One Health approach recognizes the interconnected well-being of humans, animals, plants, and their shared environment to achieve optimal health outcomes for all [Mackenzie et al., 2013]. Emphasizing a holistic strategy, it supports the development of policies and practices that address the complex interplay between ecological health and disease dynamics [Chen et al., 2024].

Plant health monitoring is crucial for ensuring the stability and security of global food supplies and maintaining the health of ecosystems [Ristaino et al., 2021]. As a fundamental component of food production, the health of plants supports biodiversity, regulates environmental conditions, and sustains agricultural productivity. The adverse effects of compromised plant health reach beyond immediate agricultural losses, affecting food security, market stability, and long term sustainability. With the growing challenges posed by plant diseases, pests, and environmental stressors, robust strategies are necessary to protect plant ecosystems, increasingly threatened by climate change and global trade [Morris et al., 2022].

Effective monitoring and management of plant health are critical, yet challenging, due to the complexity of agricultural ecosystems and the nuanced nature of threats that plants face [Bouri et al., 2023]. Accurate and timely information on plant diseases, pest infestations, and environmental conditions is essential for devising effective management and control strategies that prevent widespread agricultural crises. Traditional methods of plant health assessment often require extensive manual effort and are constrained by the scale at which they can be applied [Martinelli et al., 2015]. Consequently, there is an increasing reliance on technological advancements to streamline and enhance the accuracy of plant health monitoring.

This thesis is part of the BEYOND[1] project, which aims to improve epidemiological surveillance strategies for plant health. In this context, our research involves automatically extracting information from textual data in order to identify key factors that influence the emergence of plant diseases. We specifically aim to identify the plants involved, their pathogens, the diseases affecting them and the locations of these occurrences. This information is crucial to predict and manage the spread of diseases over the medium term [Soubeyrand et al., 2024] and also for the robustness and reliability of surveillance systems that will be used by experts. Knowing which

[1] https://beyond.paca.hub.inrae.fr/

plants are affected by specific pathogens and diseases, and their locations, allows for the construction of detailed models that can predict how these diseases might spread under various conditions. These models use historical and real-time data to simulate potential outbreaks and their progression patterns based on environmental factors, host availability, and pathogen characteristics [Bischoff et al., 2021].

1.2 . Motivation and Challenges in Named Entity Recognition for Plant Health

The information important to monitor plant health is often found in texts from places like social media and online reports. Due to the high variability and unstructured nature of this content, advanced technological tools are essential for efficient extraction. Among the technological advancements, Natural Language Processing (NLP), particularly Information Extraction techniques, offer promising tools, including Machine Learning models, for extracting actionable insights from vast amounts of unstructured textual data related to plant health. Among these tools, Named Entity Recognition (NER) is particularly significant.

Efficiently designed NER systems can automate the monitoring of vast amounts of agricultural data, significantly reducing the time and labor traditionally required for manual information scanning. For instance, by automatically detecting and extracting mentions of specific disease symptoms from new research publications or field reports, these systems facilitate the rapid updating of databases and keep key players informed about emerging threats. Furthermore, identifying the exact locations of disease outbreaks and the plant species involved are essential steps in developing tailored strategies to prevent the spread of diseases.

NER systems are designed to identify and classify specific spans of text, known as named entities, into predefined categories such as names of people, organizations, date, disease, and more. Named entities are terms that refer to real-world objects or phenomena and help distinguish these specific objects or phenomena from other similar ones (see Figure 2.10). Modern NER systems are based on Language Models, which are algorithms, that calculate the probability of a lexical unit, such as word, based on its context, and perform subsequent tasks. Language Models use various linguistic and contextual clues to determine these entities within a text, distinguishing them from non-relevant text passages through complex algorithms that integrate an implicit syntax and semantic analysis [Piantadosi, 2023]. These systems treat vast amount of texts to find specific information, serving as the cornerstone for converting text into data that can be easily managed and analyzed. This capability enhances data usability and supports various applications

across different fields, making NER a critical component in the realm of text understanding and information extraction. Consequently, Language Models are predominantly used in modern NER approaches, as well as in broader modern NLP applications.

However, when applied to specialized domains such as plant health, standard NER models frequently encounter difficulties. These models are typically designed and trained using general language data, which does not always encompass the unique vocabulary and contextual nuances specific to domains. For instance, terms like "Xanthomonas", "Phytophthora", or "Xylella fastidiosa" are commonly used in agricultural contexts to refer to specific plant pathogens, but these might not be recognized by conventional NER systems trained on standard datasets. Furthermore, the contextual significance of such terms—understanding whether "rust" refers to a fungal disease affecting plants or merely to oxidized metal—poses additional challenges that standard NER models are often incapable of handling.

In addition, the linguistic cues and patterns used to identify named entities in general texts (such as capitalization in the case of proper nouns) may differ or be less apparent in technical or scientific texts. Additionally, plant health literature might use a variety of descriptors and synonyms for a single concept, increasing the complexity of entity recognition. For example, the Asian citrus psyllid, which transmits citrus greening disease, can be named as Asian citrus psyllid, ACP, Diaphorina citri, Citrus greening spreader, Huanglongbing vector, Citrus pest, psyllid species, etc. As a result, the performance of NER systems can significantly degrade when they are tasked with processing text from specialized domains without prior adaptation or retraining.

These challenges are further intensified by the scarcity of datasets specifically designed for health monitoring across different domains (plants, human or animals) and by the diversity of textual data sources to take into account, including scientific publications, news articles, social media and other. In broader fields, abundant datasets provide a rich array of examples that enable machine learning models to predict accurately. However, in specialized fields like plant health, the data is not only scarce but also often fragmented, covering only certain aspects of the domain and lacking the comprehensive scope needed to encompass all potential scenarios and terminologies. Moreover, the nature of plant health-related data, which often includes complex descriptions of symptoms, treatments, and biological interactions, requires both detailed and comprehensively annotated datasets for effective training and evaluation of methods. Without access to extensive and diverse datasets, the models may not experience enough diverse examples to develop an accurate understanding of the domain, potentially compromising their effectiveness and accuracy.

In this context, enhancing NER systems to efficiently process and extract relevant information from agricultural texts becomes crucial. Such systems can identify in real time critical information such as the symptoms of plant diseases, locations of outbreaks, and affected plant species. This information is crucial for early warning systems and strategic response planning.

1.3 . Research Objectives and Questions

This thesis is dedicated to advancing Named Entity Recognition systems within the specialized domain of plant health, though the methodologies could also be applied to other domains. It is guided by the principle of incorporating domain-specific semantics to adapt automatic methods for recognizing named entities. It is structured around three primary objectives, each aiming to address distinct but interconnected challenges within the realm of Natural Language Processing as applied to agricultural texts. To navigate through these challenges, the study poses several research questions aimed at both testing theoretical frameworks and evaluating practical applications essential for enhancing NER capabilities in this domain:

1. The first objective aims to adapt a Language Model specifically for the plant health domain, focusing on its application in NER. The research questions for this objective explore the methods by which a Language Model can be adjusted to take into consideration the specialized terminology and contextual nuances of plant health. This includes exploring various adaptation techniques to ensure the model not only captures the domain-specific language but also excels in identifying and classifying named entities within this field. The resulting model will be then applied to the Plant Health domain and evaluated as detailed in objective 3.

2. The second objective seeks to develop a method enabling an existing NER system to recognize new named entities without the necessity for explicit fine-tuning. The questions here delve into the potential of using unsupervised, semi-supervised learning methods or zero-shot techniques to enhance the adaptability of NER systems. These questions seek to uncover how these systems can autonomously recognize and categorize new entities based on linguistic patterns and features found in unstructured text data, thereby reducing reliance on extensive annotated datasets. The developed system will be further used in Plant Health domain as outlined in objective 3.

3. The third objective focuses on designing a comprehensive adaptation system that is specifically tuned for the plant health domain, and built

upon the findings from the first two objectives. Research question for this objective addresses whether recombining different modules of the same system, each fine-tuned separately for similar tasks, can achieve results comparable to those of a classically fine-tuned model. This process entails combining the adapted Language Model with introduced entity recognition strategies into a robust, scalable system capable of managing the complex data scenarios typical in plant health monitoring.

Together, these research questions frame a study that seeks to extend the boundaries of current NER technologies and provides a focused approach to the real-world challenges of the plant health sector. By addressing these questions, the research aims to contribute to the field of natural language processing by providing advanced tools that can improve the monitoring and management of plant health globally.

1.4 . Methodological Approach

The methodology used in this thesis leverages advanced machine learning techniques, particularly adapting a Language Model to a domain (before adapting it to NER) with proposed KeyWord Masking technique and the integration of entity type semantics into Language Models. Namely, we are interested in determining whether selecting specific key words for our model to focus on could enhance the adjustment process.

Another problem that we are interested in is minimizing the amount of training data needed to adapt a Language Model to a particular task. To accomplish this, we use the latent representations of entity types. These representations are derived from documents that focus on topics related to specific entity types and are inserted into a NER model as input features. This integration allows the NER system to benefit from a semantic understanding of entity types, even in the absence of large annotated datasets. By embedding these latent representations into the classifier, the system can use inherent semantic and contextual cues from the data to recognize and categorize new entity types. This approach minimizes the dependency on extensive manual annotations, which are often costly and time-consuming to produce.

Both strategies are designed to enhance the model's broad understanding of text and entity relationships, enhancing the adaptability and transferability of the NER systems developed. We evaluate our approaches on diverse domains, including Plant Health. This focus on detailed semantics ensures that the system is versatile and capable of handling diverse and complex data scenarios that are typical in specialized domains.

1.5 . Thesis Structure

The thesis is organized into several chapters, each focusing on distinct aspects of the research:

- **Chapter 2: Background and Related Work** – This chapter discusses the fundamentals of Natural Language Processing and reviews existing literature on Named Entity Recognition and domain adaptation techniques, highlighting their relevance and application in various fields, including the specifics of plant health domain.
- **Chapters 3, 4 and 5: Methodology and Experiments** – These chapters detail the methodologies developed for adapting NER systems to the specialized domain of plant health. They describe the experimental setups and discuss the results obtained from these experiments. Each chapter addresses a specific objective mentioned in the list 1.3: Chapter 3 focuses on Language Model adaptation using keywords, aligned with objective 1; Chapter 4 explores NER domain adaptation techniques, meeting objective 2; and Chapter 5 integrates these methods and applies them to the plant health domain, fulfilling objective 3.
- **Chapter 6: Conclusion** – This final chapter summarizes the key findings and contributions of the thesis. It discusses the broader implications of this research for future academic inquiries and practical applications, and suggests potential avenues for further research.

2 - Background and related work

Language models have transformed the field of Natural Language Processing (NLP), enabling machines to accomplish tasks requiring comprehension, interpretation, and generation of human language. From early n-gram models to recent transformer-based architectures such as BERT [Devlin et al., 2019] and GPT [Radford et al., 2018], language models have consistently expanded the possibilities in understanding and generating natural language. Particularly, these advancements have proven instrumental in enhancing Information Extraction techniques, which is the primary focus of this thesis.

This chapter provides a thorough overview of the fundamental principles, key models, and notable advancements that have influenced the current landscape of language modeling and NER applications. It begins by establishing essential concepts crucial for subsequent section understanding, covering basic machine learning and NLP principles.

2.1 . Fundamentals of Machine Learning for NLP

Machine learning principles form the foundation of recent advancements in natural language processing. The evolution of language models from early statistical methods to sophisticated neural architectures underscores the pivotal role of ML in modern NLP. Before exploring modern language models, it is important to establish a strong understanding of core machine learning concepts. This subsection provides a primer on these foundational principles, laying the groundwork to delve into more advanced models and their applications in NLP, particularly in NER.

In Machine Learning, the notion of a "task" refers to a specific problem that the model is designed to solve. Below are some main types of tasks [Sarker, 2021]:

- **Classification tasks** involve predicting a category or label for input data from a predefined set of categories or labels (e.g., language identification).
- **Regression tasks** consist in predicting a continuous value based on input data (e.g., predicting temperature based on meteorological data).
- **Clustering tasks** aim to group a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups (e.g., organizing documents into groups based on topic similarity).

- **Dimensionality Reduction tasks** imply simplifying the input data, generalizing the most relevant features (e.g., simplifying the feature set of audio data).

- **Reinforcement Learning tasks** focus on learning a strategy to maximize the reward an agent receives for its actions in a given environment (e.g., autonomous driving).

Named Entity Recognition, which is central to this thesis, is usually formulated as a classification task [Hu et al., 2024] where the objective is to predict a category for each segment of text, such as individual words, from a set of predefined labels, such as "Location", "Pest", etc. (see Section 2.3). This process of categorization is a fundamental application of supervised learning principles, where the model learns to associate specific features of the text (input) with the correct labels (output). This task is primarily solved with supervised learning approaches, which will be described in the subsequent section.

2.1.1 . Supervised Learning

Supervised learning is a type of machine learning approach focused on finding a mapping between input features and outputs [Cunningham et al., 2008], enabling accurate predictions for new, unseen data. This requires a dataset, or labeled data, where each input sample is matched with the corresponding output sample, or label, that model aims to predict afterwards. The dataset must be collected, inspected and annotated in advance. Supervised learning is widely used for most NLP tasks [Sarker, 2021], including Named Entity Recognition.

Feed-Forward Neural Network

A neural network is a mathematical model for data processing inspired by the architecture of the human brain (hence the name). The classic feed-forward neural network was introduced in [Rosenblatt, 1958] to understand the complex relationships in data through computational means. This foundational concept was further validated by the universal approximation theorem [Hornik et al., 1989] which claims that such a network with at least one hidden layer can approximate any continuous function with any level of precision, given sufficiently many neurons in its hidden layer. Typically, this network is composed of several layers, each containing multiple neurons that process incoming data sequentially from the input to the output layer. A neural network is classified as "deep" when it contains at least five hidden layers [LeCun et al., 2015], allowing it to capture more complex hierarchies of information.

A basic computational unit of any type of neural network is a neuron (see figure 2.1). It processes inputs, which are weighted outputs of the previous layer or external inputs. The operation within a neuron involves summing these weighted inputs, adding a bias, and then passing the result through an activation function. Weights are parameters that determine how much each input is important for this neuron. The bias is a parameter which ensures the activation of the neuron even if all inputs are zero and adjusts the activation function to adjust the neuron's output. The final goal of training is to find the optimal parameters. The activation function is crucial as it acts as a filter, determining the extent of which a current neuron's output should be considered. Depending on its nature, it also can introduce non-linear properties to the model, enabling it to learn complex patterns and behaviors that simple linear equations cannot.

While each neuron operates based on its inputs and activation function, it is integrated into a larger network where the patterns of connectivity significantly enhance the learning capabilities. It is important to note that the activation function is identical for all neurons in the same layer. This uniformity ensures consistent processing of signals across that layer, contributing to the network's ability to generalize from input data to form outputs. The final output of the network is a vector with a number of coordinates equal to the number of neurons in the last layer. Thus, an output value of a j-th neuron is calculated by the following formula:

$$y_j = F\left(\sum_{i=1}^{n} w_{ij}x_i + b_j\right), \tag{2.1}$$

where x_i is the output from the i-th neuron in the previous layer, w_{ij} denotes the weight applied to this output by the j-th neuron, b_j is a bias and F is the activation function that processes the weighted sum, adjusted by the bias, to produce the output y_j.

The output of each neuron is then forwarded as input to neurons in the subsequent layer, and this process repeats across the network (see Figure 2.2). The final layer's output is typically transformed into a format suitable for addressing specific types of problems, such as classification or regression. For example, in a classification task, the final layer may use a softmax function to convert the outputs into probability distribution, representing the model's confidence in each class.

Feed-forward neural networks, like other types of neural networks, are trained using backpropagation [Rojas and Rojas, 1996], a method where errors in predictions are used to iteratively adjust the model's parameters. It begins with the calculation of the loss function, which measures the difference between model's current output and the desired output. The next step involves determining how each parameter impacts on the loss

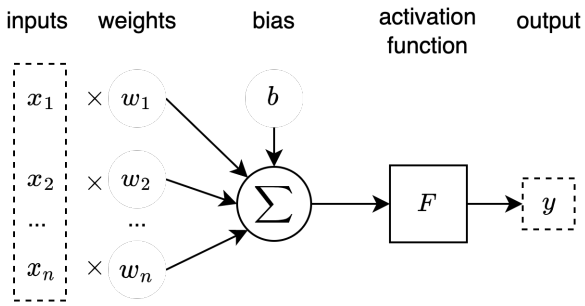

Figure 2.1: **Schematic Representation of a Single Neuron in a Neural Network.** This diagram depicts a single neuron where input values $x_1, x_2, \ldots, x_n$ are each multiplied by corresponding weights $w_1, w_2, \ldots, w_n$. The weighted inputs are then summed to produce the output y, demonstrating the fundamental operation within a neural network neuron.

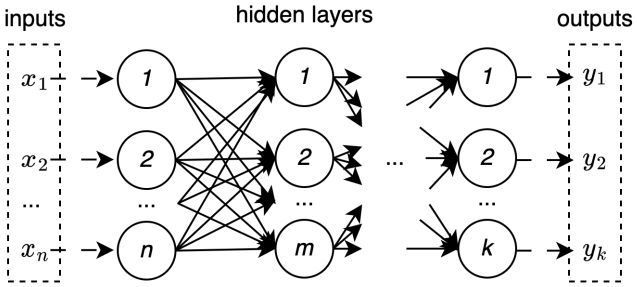

Figure 2.2: **Feed-Forward Neural Network.** This diagram illustrates a standard neural network consisting of multiple layers. Each layer is depicted with a set number of neurons, represented as circles. The first layer contains n neurons, the second layer m neurons, and the final layer k neurons, showcasing the structure of a typical feed-forward network architecture.

function and then adjusting these parameters to minimize the error. These adjustments are made using a technique called gradient descent [Amari, 1993]. Backpropagation improves computational efficiency by computing gradients for each layer using only the gradients from the subsequent layer. Starting from the output layer, the method calculates what is sometimes referred to as the local error for each neuron — the neuron's specific contribution to the total network error. These local errors are then used to compute gradients for previous layers, propagating the error back through the network, hence the name 'backpropagation'. This iterative process continues until the network's predictions are sufficiently accurate or meet a predefined criterion of accuracy.

Thus, by leveraging layers of neurons with the ability to learn from data, feed-forward neural networks form powerful tools for a wide range of data processing tasks, including Named Entity Recognition [Xu and Wang, 2021].

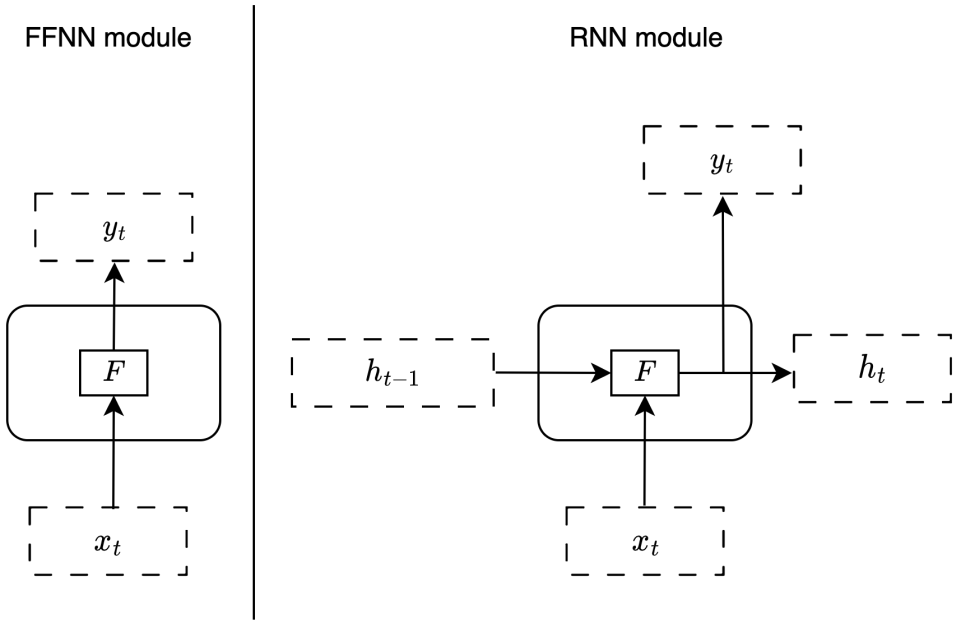

Figure 2.3: **Comparison of Hidden Layers in FFNNs and RNNs.** This figure illustrates the key differences in network architecture between Feed-Forward Neural Networks (FFNNs) and Recurrent Neural Networks (RNNs). In the left panel, an FFNN module is depicted where connections between layers propagate strictly forward without any feedback loops; x_t represents the input at time step t, and y_t denotes the output or hidden state at the same step. The right panel illustrates an RNN module, highlighting its recurrent connections that allow feedback from the network's previous output states (h_{t-1})sequences. This setup enables the RNN to maintain a continuous flow of information, where x_t is the input, y_t is the output and h_t is the hidden state at time t.

Nonetheless, subsequent sections will introduce more efficient models that offer advanced capabilities and improved performance for such applications.

Recurrent Neural Networks

Recurrent neural networks (RNNs), introduced by [Amari, 1972], share a basic architectural similarity with feed-forward networks but incorporate a crucial modification for processing sequential data, such as time series or text. This capability is achieved through the following architectural modification: each neuron can receive input not only from the preceding layer but also from its own previous output (see Figure 2.3) through a mechanism known as a hidden state. The concept of a *time step* within this hidden state corresponds to each discrete point in the sequence (a letter or a word in text, or a moment in time). This allows the model to maintain and process context from one segment to the next, adapting dynamically to the sequential input. The hidden state in an RNN can be represented by the formula:

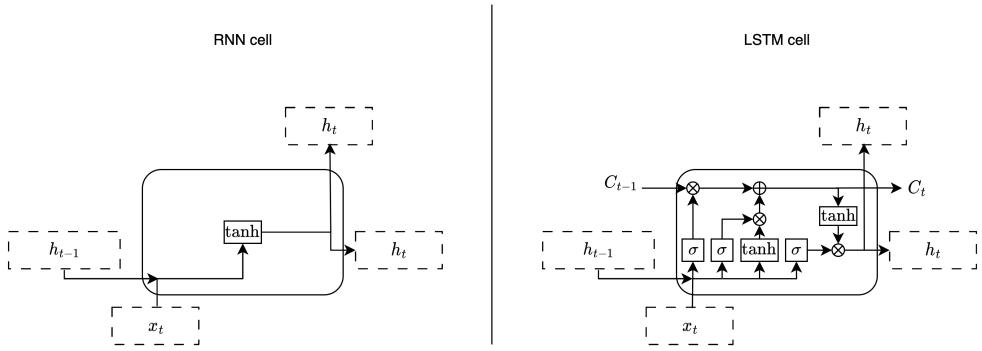

Figure 2.4: **Comparison of RNN and LSTM Cell Architectures.** This diagram contrasts the internal structure and processing flow within a standard Recurrent Neural Network (RNN) cell and a Long Short-Term Memory (LSTM) cell. It details the simple recurrent loop of the RNN cell alongside the complex gate mechanisms of the LSTM.

$$h_t = F(Ux_t + Wh_{t-1} + b)$$

where h_t is the hidden state at time t, x_t is the input at time t, and F is the activation function, b is the bias. The matrices U and W are the weight matrices for the input and the recurrent connection, respectively.

This architecture effectively creates a form of "memory" that retains information about previous inputs, enabling the network to make predictions that consider the sequence's history. Due to this capability, RNNs perform better then FFNNs in tasks involving sequential data, such as text generation, machine translation, and Named Entity Recognition [Ali et al., 2022, Li et al., 2015b, Li et al., 2015a].

Long Short-Term Memory Neural Networks

Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber in 1997 [Hochreiter and Schmidhuber, 1997], represent a specialized type of RNN designed to capture long-term dependencies. An LSTM network consists of units known as cells, each containing memory and gating mechanisms crucial for regulating information flow and retention. Unlike RNNs that use a single operation in each hidden unit, LSTMs incorporate multiple gates within each cell to enhance memory processing. In an LSTM cell, a gate is a component that acts as a selective filter: it decides whether to retain or forget input information based on its relevance to the current task, thus modeling a short-term memory. Another key component of LSTM is the cell state, which undergoes only a few linear transformations, preserving the information over time and thus modeling a long-term memory. This mechanism allows controlling the information flow (see Figure 2.4) and

enables the network to selectively retain or forget data. The operations within these gates are described as follows:

First, the forget gate determines which information is retained or forgotten by the network:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

where W_f is the weight matrix, $[h_{t-1}, x_t]$ is the concatenation of the previous hidden state and the current input, b_f is the bias, and σ represents the sigmoid activation function.

Then, the input gate decides which values are important to update:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

where i_t is the input gate activation at time t. W_i is the weight matrix, and b_i is the bias for the input gate.

The network also creates a vector of new candidate values that could be added to the state of the cell:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

where $\tilde{C}_t$ represents the candidate values for updating the cell state. W_C is the weight matrix, and b_C is the bias, with $\tanh$ indicating the hyperbolic tangent function.

The cell state is then updated as follows:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

where f_t is the activation of the forget gate at time t, which moderates the retention of the previous cell state C_{t-1}, and i_t is the activation of the input gate at time t, controlling the degree of inclusion of new candidate values $\tilde{C}_t$ into the cell state.

Finally, the cell output is computed by combining the cell state with the hidden state:

$$h_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) * \tanh(C_t)$$

where h_t is the hidden state at time t, W_o is the weight matrix, b_o is the bias for the output gate, and C_t is the updated cell state. This output forms the basis for subsequent operations within the network.

There are several variations of LSTM. One important variation is the Bidirectional LSTM (BiLSTM), introduced by [Zhang et al., 2015]. It comprises two LSTMs layers, one of which processes the input in a backward direction. This dual mechanism allows the model to consider both past and future contexts simultaneously (see Figure 2.5), enhancing its predictive accuracy across various sequence modeling tasks including NER.

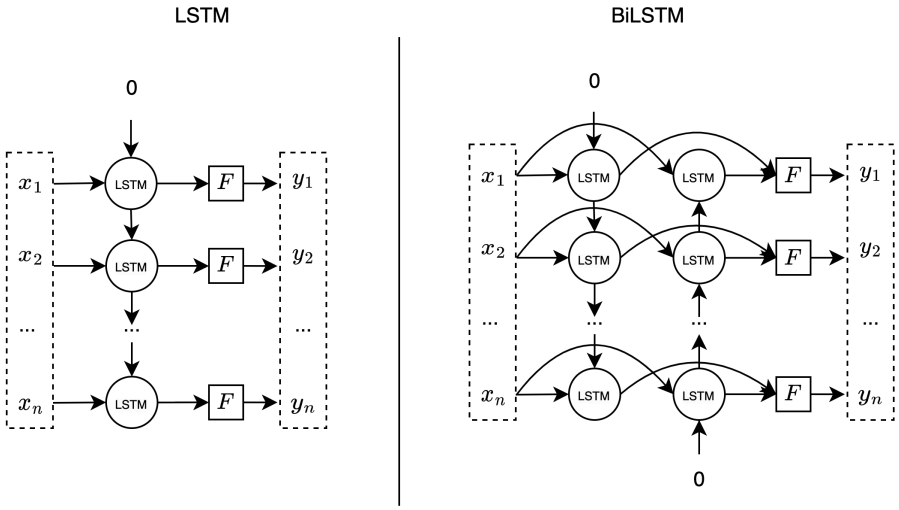

Figure 2.5: **Comparison of LSTM and BiLSTM Architectures.** This diagram illustrates the architectural differences between Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) networks. On the left, the LSTM processes input sequences (x_1, x_2, ..., x_n) in a forward direction, with each LSTM unit outputting a corresponding y value (y_1, y_2, ..., y_n) after passing through an activation function F. On the right, the BiLSTM processes input sequences both forward and backward, allowing the network to incorporate information from both past and future contexts within the sequence. Each unit in the BiLSTM consists of two LSTMs, one for each direction, whose outputs are combined at each time step to produce the final output.

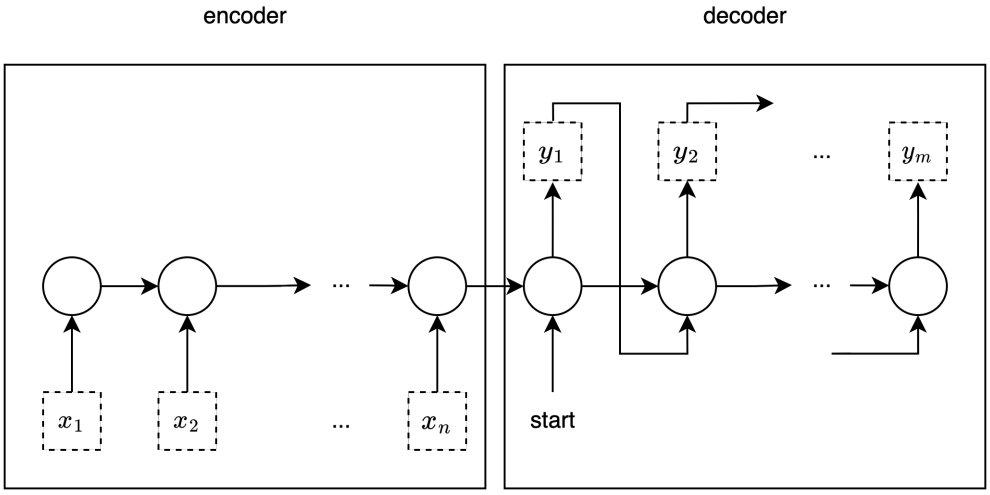

Figure 2.6: **Sequence to Sequence model architecture.** The lower row represents the input sequence $x_1, x_2, \ldots, x_n$, processed by the encoder. The upper row represents the output sequence $y_1, y_2, ..., xy_m$. The circles in the middle denote the network units that can be network units, such as RNN or LSTM units as described in Sections 2.1.1 and 2.1.1 respectively.

Overall, LSTMs provide a robust architecture for managing long-term dependencies, an important capability for complex sequence modeling tasks across various domains, including NER [Lample et al., 2016, Limsopatham and Collier, 2016, Zhai et al., 2018].

Sequence-to-sequence

The Sequence to Sequence (seq2seq) architecture, introduced by [Cho et al., 2014], was originally developed for machine translation tasks. It is designed to transform a sequence (whether of letters, words, or even images) into another sequence via an intermediate representation. This architecture consists of two main components: an encoder and a decoder. The encoder processes the input sequence into a latent representation, or a context vector, which is a compressed numerical representation that captures all the essential information from the input. The decoder then incrementally constructs the output sequence. Specifically, each unit of the decoder produces a segment of the output (such as a letter or a word) based on the output from the previous hidden state, starting with the context vector. The schema of this architecture is presented in Figure 2.6.

In the original paper [Cho et al., 2014], both the encoder and decoder are implemented using LSTM networks. However, various architectures can be employed depending on the specific application. For instance, in Named Entity Recognition, modifications of LSTM networks combined

with Conditional Random Fields (see section 2.1.1) have shown effectiveness [Zhu et al., 2020, Chen and Moschitti, 2018].

Although, seq2seq was initially designed for machine translation tasks, it has proven more efficient for Named Entity Recognition than the previously discussed architectures [Tan et al., 2021, Chen and Moschitti, 2018, Wang et al., 2019]. It has additionally inspired the design of the Transformers architecture which continues to set the standard for many tasks, as will be detailed in the subsequent section.

Transformer

The Transformer Network, introduced in [Vaswani et al., 2017], builds on the foundational ideas of the seq2seq model but uniquely implements a "multi-head attention" mechanism (see Figure 2.7), which will be defined and explained later in this section. This architecture was created particularly for text treatment. Unlike sequence-to-sequence networks that rely heavily on recurrent layers, the Transformer consists of an encoder and a decoder. The encoder processes the input information (e.g., a text) and transforms it into a latent feature representation. The decoder then reconstructs this representation into a new sequence (e.g., an answer to a question, a translated sentence, etc.). Each of the encoder's six layers is a feed-forward network equipped with a multi-head attention mechanism, enhancing its ability to focus on different parts of the input sequence simultaneously. The decoder's structure mirrors that of the encoder but includes an additional masked multi-head attention sub-module in each layer to ensure that each part of the output sequence does not prematurely influence the parts that come after it.

In the context of NLP, a transformer network processes input data through what is known as an input embedding, which consists of three main components: a token embedding, a segment embedding, and a positional embedding. A token represents a unit of language, such as a letter, a word, or a sentence. While using transformers, a token is defined as a part of a word, derived from a predefined vocabulary using a WordPiece algorithm [Wu et al., 2016]. Token embeddings transform text into these tokens and assign a unique identifier to each from the vocabulary. Segment embeddings differentiate various parts of the input text, usually by marking the current segment being processed with ones and the rest with zeros. Lastly, positional embeddings record the position of each token within the text to maintain sequence information.

The attention mechanism enables the model to focus on each unit of the input separately when processing. Specifically, the sequence of token embeddings is represented as a matrix X, where each row corresponds to a token embedding. This matrix X is then linearly projected by multiplying

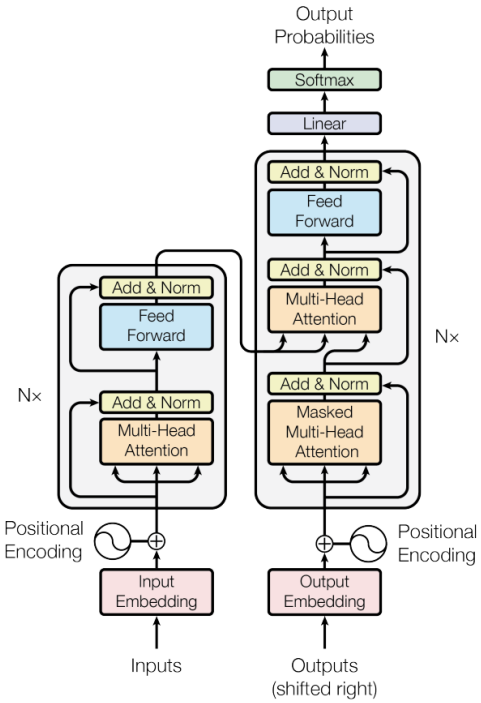

Figure 2.7: **Overview of the Transformer Model Architecture.** This schema, taken from the original paper [Vaswani et al., 2017], illustrates the main components of the Transformer model, highlighting both the encoder and decoder structures. Each module consists of multiple layers repeated N times. The encoder processes input embeddings, enhanced with positional encoding, through multi-head attention and feed-forward networks. The decoder follows a similar architecture but incorporates an additional masked multi-head attention layer to prevent the influence of future context on the predictions.

it with parameter matrices to form three distinct representations: query (Q), key (K), and value (V) matrices. The query matrix represents the relevance of each token in relation to others from its own perspective. Conversely, the key matrix indicates the relevance of each specific token from the perspective of other tokens. The value matrix is a latent representation of each token calculated from its occurrence in a context with other tokens. These matrices are produced by multiplying the input embeddings with corresponding weight matrices. These transformed matrices are then used as inputs to the scaled dot-product attention function, which calculates a weighted latent representation across all tokens:

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

where d_k represents the dimensionality of the key vectors, which is used to scale the dot product, thereby assisting in maintaining stable gradients during

training.

The multi-head mechanism (see Figure 2.8) implies a concatenation of multiple attention functions operations, each with its own set of learned weights. By segmenting the query, key, and value matrices into multiple "heads", the model can simultaneously have access to information from multiple positions and representations angles. This diversification enhances the model's ability to focus on various parts of the input sequence and extract a richer set of features.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O$$

where each head_i is computed as:

$$\text{head}_i = \text{Attention}(Q, K, V) = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V)$$

In this configuration, $W_i^Q, W_i^K,$, W_i^V, and W^O are the weight matrices for the query, key, value and the output linear transformation for the ith attention head, respectively. This structure enables the model to capture different types of relationships in the data, making it adept at modeling complex dependencies across extended sequences.

The masked multi-head attention mechanism is very similar, but during training, a mask is applied to the attention mechanism to hide the information about future tokens in the sequence. Specifically, the attention weights corresponding to future tokens are set to a very large negative value (e.g., negative infinity) before applying the softmax function. As a result, these attention weights effectively become zero after applying the softmax function, meaning that the model completely ignores information about future tokens during training. This helps the model learn to generate outputs sequentially and autoregressively, preserving the order of tokens and maintaining grammatical coherence and semantic accuracy in the generated sequences.

Thus, Transformer architecture is a significant advancement in processing sequential data, including text, effectively capturing dependencies across long sequences while maintaining the order and coherence of generated outputs. Widely regarded as state-of-the-art, Transformer and its variants continue to excel across various tasks, including Named Entity Recognition [Yang et al., 2024, Yan et al., 2019, Lothritz et al., 2020].

Conditional Random Fields

Conditional Random Fields (CRF), introduced in [Lafferty et al., 2001], are frequently used to refine the outputs of a neural networks. CRF is a special type of Markov Random Field (MRF) (see Figure 2.9), characterized by an undirected graph where random variables (representing both inputs and

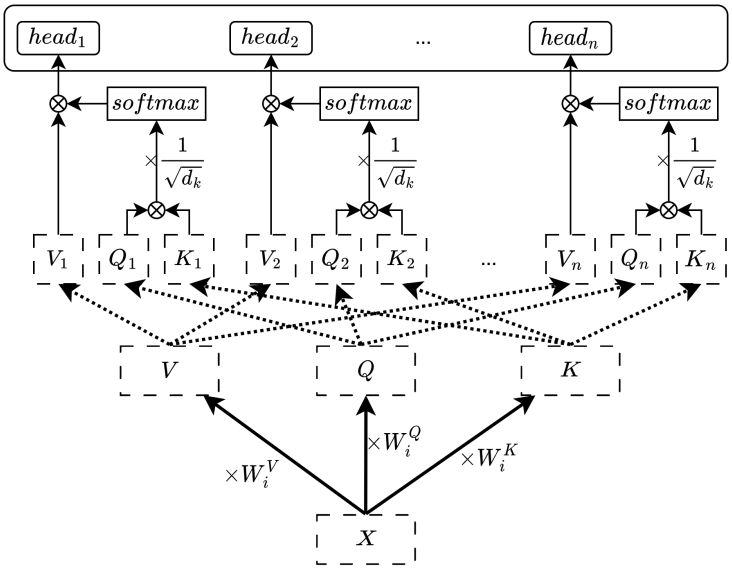

Figure 2.8: **Multi-Head Attention Mechanism in a Transformer Model.** This diagram depicts the multi-head attention mechanism, where each "head" processes the input data independently. Bold arrows represent linear transformations, applying a unique set of weights to the input. Dashed arrows indicate the extraction of specific rows from the matrix without any transformation. After processing through a softmax layer to calculate attention weights, the outputs from all heads are concatenated, forming a comprehensive final output that enhances the model's ability to focus on various aspects of the input data simultaneously.

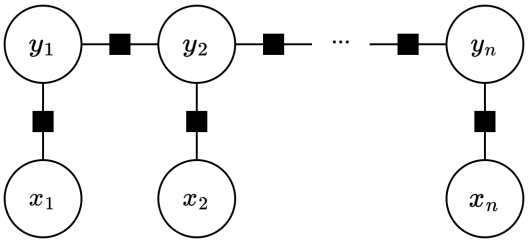

Figure 2.9: **Illustration of a Conditional Random Field Model.** This figure depicts the basic structure of a CRF model, where x_i and y_i represent the input and output variable nodes, respectively, for each position in a sequence from 1 to n. The black squares in the diagram denote the transition feature function f, as outlined in the formulas presented in this section.

outputs in our context) act as edges, and dependencies between these variables act as vertices, all adhering to Markov properties [Matúš, 1992]. Unlike general MRFs, CRFs include a set of functions, each corresponding to a complete subgraph (or clique), which assign non-negative real values to every potential state of the subgraph's elements. Specifically, CRFs compute the conditional probabilities of output variables based on input variables. More precisely, the conditional probability is given by:

$$P(y|x) = \frac{1}{Z(x)} \prod_k \exp\left(\sum_k \lambda_k f_k(y, y_{-1}, x_t)\right)$$

where k indexes the cliques in the graph, $\phi_k(x_{\{k\}})$ is a potential function describing the state of random variables in the k-th clique, f_k is a feature function associated with the k-th clique, λ_k are the parameters learned during the training and Z is a normalization factor, calculated by:

$$Z(x) = \sum_y \prod_k \exp\left(\sum_k \lambda_k f_k(y, y_{-1}, x_t)\right)$$

where y is a set of possible output sequences.

Thus, CRFs explicitly model the relationships between consecutive labels in a sequence. This structure allows CRFs to effectively handle the conditional dependencies between observations and their labels, enhancing prediction accuracy for complex sequences. In the context of Named Entity Recognition, a CRF layer is often applied on top of another model, such as an LSTM or a Transformer. In this setup, x_i represents the features of the i-th word, while y_i denotes the predicted class for that word. This provides a more nuanced understanding and yields better predictions than using a traditional sequence model alone [Lample et al., 2016, Ma and Hovy, 2016].

This section has explored various supervised learning architectures, each offering significant advantages for handling specific types of data and

tasks within the realm of NLP, particularly for NER. However, there are other supervised learning architectures and techniques not discussed here, primarily because they may not provide additional context that significantly diverges from what has already been examined or may not be as relevant for our subject.

2.1.2 . Unsupervised and semi-supervised methods

Beyond supervised methods, which rely on pre-labeled data, unsupervised and semi-supervised methods are also employed. Unsupervised learning aims to discern patterns directly from data, providing a valuable perspective in situations where labeled data is scarce or costly to acquire. However, these results are typically less accurate. Additionally, unsupervised algorithms are more sensitive to scalability issues and data noise. Unsupervised algorithms are predominantly used when no labeled data is available. Conversely, if minimal labeled data exists, semi-supervised algorithms offer a good balance. These algorithms are trained using a mix of labeled and unlabeled data, but their performance is generally worse than fully supervised approaches and at the same time still require some labeled data. A review of unsupervised and semi-supervised algorithms applied to NER will be presented in Sections 2.3.5 and 2.3.5 respectively.

2.1.3 . Few-shot Learning

In addition to supervised, unsupervised and semi-supervised methods, few-shot learning represents another approach that operates with a limited amount of data. Few-shot learning is a generic name of ML methods that use only a small number of examples for model adjustment. This is often confused with semi-supervised learning due to its use of few labeled samples. However, few-shot learning specifically refers to using a small set of examples for training without implying the nature of the learning algorithm itself. In contrast, semi-supervised methods involve combining both labeled and unlabeled data during training and do not specify the number of examples used. Nonetheless, semi-supervised approaches are frequently evaluated in few-shot contexts due to their reliance on limited labeled instances. Understanding this distinction is crucial for grasping the practical applications and limitations of each method in scenarios with scarce data. This thesis explores a zero-shot technique that operates under a supervised framework but is applied to data that has not been previously used by the model, nor is it similar to any data used while constructing the algorithm. This approach is called zero-shot strategy. Due to the infrequent evaluation of supervised methods in this context, we will also compare our approach to unsupervised and prompt engineering methods.

2.1.4 . Adaptive Learning Methods

When we would like to adjust a model without extensive retraining, we need to use adaptation techniques. These techniques fall into two primary categories: transfer learning and meta-learning.

The meta-learning approach involves training a model on a variety of tasks, then adapting it to perform a specific target task with minimal examples. Conversely, transfer learning implies using a model that is pre-trained on a large dataset for one single task or domain, which is then fine-tuned on a smaller, specific target dataset somehow similar to the dataset used for pre-training. In this work, we will focus on employing a transfer learning strategy.

As we turn now to the domain of Language Models, we will delve into how these advanced tools build on and differ from the supervised techniques outlined, continuing our exploration of their pivotal role in NLP advancements.

2.2 . Language Modeling

A model is a simplified representation of an object, system, or process designed to facilitate the analysis, understanding, and prediction of its specific aspects. Language Models (LMs) are designed to predict the probability of a sequence of words or to generate new text sequences that are both syntactically consistent and contextually meaningful.

Historically, the field has evolved from simple n-gram models, which rely on the frequencies of short sequences of words, to complex deep learning models like transformers [Vaswani et al., 2017], which are capable of capturing long-range dependencies and subtle nuances in text. The advent of models such as GPT [Radford et al., 2018] (Generative Pretrained Transformer) and BERT [Devlin et al., 2019] (Bidirectional Encoder Representations from Transformers) has significantly advanced the capabilities of Language Models. Today, these models underpin a multitude of NLP applications propelling them towards achieving unparalleled accuracy in both understanding and generating human language.

2.2.1 . Language Models pre-training and fine-tuning

Language Models are typically used in various NLP task, such as automatic translation, text classification, information extraction, etc. Before these models are adjusted for specific applications, they must first undergo a pre-training stage. This stage equips them with foundational language knowledge, typically through an unsupervised learning task that simulate language generation. A common approach to this is Causal Language Modeling (CLM), which involves predicting a token based on the preceding sequence of tokens. Conversely, Masked Language Modeling (MLM) implies

replacing certain tokens with a special "mask" token and then predicting the original tokens from this masked input. The MLM task was introduced in [Devlin et al., 2019] with a development of a Language Model named BERT (see Section 2.2.4), where 15% of the tokens in each sequence are masked. After this pre-training stage, language models are adjusted to optimize their performance for particular tasks. This process, known as fine-tuning, involves modifying the model's parameters to better align with the characteristics and requirements of the target task. Often, this adjustment includes a second round of training on a domain-specific dataset, ensuring that the model captures the nuances and complexities of the particular domain of texts it will be applied to. Thus, the pre-training stage of Language Modeling is a crucial component of modern language models, enabling them to learn rich contextual representations, perform well on a wide range of NLP tasks, and leverage transfer learning effectively.

2.2.2 . Early Developments

While the specific term "Language Models" may not have been coined until the later part of the 20th century, the pursuit of understanding and generating human language through computational means has a rich history. Early efforts in this domain leveraged basic statistical methods and feature extraction techniques, emphasizing the frequency and significance of words within text data (e.g., co-ocurence matrices [Leydesdorff and Vaughan, 2006]). These initial approaches laid the groundwork for what would eventually be recognized as language modeling.

Term Frequency (TF) quantifies how often a specific word appears within a document, which is hereinafter defined as a string of characters. This metric provides a straightforward measure of word importance. This approach operates on the premise that the more frequently a word occurs in a text, the greater its significance for the document's overall theme or content. Such simplicity made term frequency an essential initial step in text analysis, however, while it highlights key terms, Term Frequency alone does not account for the likeness of words across multiple documents, which led to the development of more nuanced approaches that consider word significance in broader contexts.

Term Frequency-Inverse Document Frequency (TF-IDF) first proposed in [Sparck Jones, 1972] improves upon the simplicity of term frequency by also considering the occurence of a word across the whole collection of documents. This dual consideration allows TF-IDF to assign higher weights to words that are frequent in a specific document but rare in the corpus, effectively distinguishing them as more relevant or unique

to the document's context. As such, TF-IDF offers a more sophisticated measure of word importance, enabling more accurate identification of a document's key themes and facilitating tasks such as document similarity comparisons and information retrieval. As a foundational vector space model for document processing [Klampanos, 2009], TF-IDF also acts as a precursor to contemporary language models. Even though TF-IDF is not directly applied to solve NER task, it is used to preprocess input for further usage by neural network models, such as Word2Vec (see Section 2.2.4).

Vector Space Models (VSMs), introduced by [Salton, 1971] and also known as term vector models, represent text documents as vectors within a geometric space. In this model, the spatial distance between any two document vectors reflects the similarity or relevance of the documents to each other. Documents are represented in a document-term matrix, which is constructed using Term Frequency or Term Frequency-Inverse Document Frequency scores. The similarity between documents can be quantified using distance measures such as Euclidean Distance or Cosine Similarity. For handling large matrices, dimensionality reduction techniques, such as Singular Value Decomposition [Deerwester et al., 1990], are often employed. VSMs are particularly effective in identifying semantic relationships between words and topics within a corpus and are widely used in applications such as topic modeling and information retrieval. As foundational components in the evolution of language processing, VSMs set the stage for the development of more complex statistical models and neural approaches that dominate current research and applications in natural language understanding.

2.2.3 . Statistical Models

Statistical models rely on the probabilities of sequences of words or linguistic elements (such as named entities) to predict and interpret new sequences. These models operate on the premise that language exhibits statistical regularities, such as the tendency of certain words to co-occur frequently or the likelihood of particular word sequences occurring within a given context. This principle is central to distributional semantics, which posits that words observed in similar contexts have similar meanings [Harris, 1954]. By capturing these patterns, statistical models can generate coherent and contextually appropriate text, or a sequence of consecutive labels such as parts of speech, named entities, etc.

N-gram models predict the probability of a token (e.g., word or character) n based on the occurrence of its preceding $(n-1)$ tokens, treating language as a Markov process [Jurafsky and Martin, 2018]. In the context of NER, such models predict the probability of both the n-th token and its corresponding

label, as proposed in [Jahangir et al., 2012]. Despite their simplicity and efficiency, n-gram models are severely limited by their inability to capture extensive contextual dependencies in text. Additionally, they rely only on observable variables and are challenged by data sparsity issues as the number of tokens increases.

Hidden Markov Models (HMMs) [Baum et al., 1970] offer a more complex approach to sequence modeling. Namely, HMMs operate on the premise of a sequence of hidden (unobservable) states that generate observable outputs. Each state has a probability distribution over potential output tokens (e.g., words or characters), and transitions between states are governed by their own probabilities. This architecture allows to capture longer and more abstract dependencies that extend beyond immediate token sequences. HMMs are particularly applicable in tasks where each observable data point (e.g., a token) is believed to be generated by a hidden state (e.g., an entity type). Despite their effectiveness in modeling contextual dependencies, HMMs are constrained by a fixed-length context window, typically short. Additionally, as the quantity of tokens within the context grows, there is an exponential increase in computational complexity due to the expansion of potential states. Despite their limitations, HMMs remain useful in some applications, but they are supplanted by more advanced neural language models capable of capturing complex linguistic phenomena with higher precision.

2.2.4 . Neural Language Models

The advent of neural networks introduced a new paradigm for language modeling, overcoming many limitations of traditional statistical models. This progress has been further accelerated by the availability of computational resources, allowing for more complex and sophisticated model training and deployment. In essence, neural language models represent a data-driven approach to language modeling, where neural networks iteratively refine their internal representations of language based on the underlying statistical properties of the data. Unlike statistical language models that rely heavily on the frequency and arrangement of words, neural models use high-dimensional, continuous embeddings that capture subtle linguistic cues and contexts. This shift enhances adaptability and scalability of models. Unlike traditional statistical approaches that rely on manually crafted features, these Language Models rely on a neural network architecture, coupled with a task definition tailored to the specific language processing objective at hand.

An early approach, introduced in [Bengio et al., 2000] was created to overcome the high-dimensionality problem of word vectors produced by

statistical models. This neural approach effectively captures semantic nuances, enabling more accurate prediction of word sequences compared to traditional n-gram models. The architecture is essentially a perceptron as described in section 2.1.1. Thus, this method introduced distributed word representations. It developed a function to predict words based on aggregated context features—namely, the distributed word vectors-, streamlining the understanding of language context.

The classical algorithm for word vector representations (embeddings) known as Word2Vec, introduced in [Mikolov et al., 2013], offers two models: Continuous Bag of Words (CBOW) and Skip-gram. In both cases, the neural network contains one hidden layer. The CBOW model uses several words from the past and several words from the future context to predict the current word using a log-linear classifier. It is efficient in processing frequent words and generally learns more quickly than Skip-gram, but it does not account for the order of words. Conversely, Skip-gram employs a single word as input to predict nearby words, using again a log-linear classifier with a continuous projection layer. This method is known for producing higher quality word vectors by considering a broader context range in its predictions but shares a common limitation with CBOW in that both models assign static vectors to words, ignoring the polysemy where words can have multiple meanings based on context. Additionally, neither model can generate embeddings for out-of-vocabulary words not seen during training, limiting their flexibility in adapting to new texts or domains where novel vocabulary might emerge.

A similar approach, known as FastText, is introduced in [Bojanowski et al., 2017], which differs from Word2vec models by treating character n-grams, rather than whole words, as both input and output data. This method enhances the model's ability to capture subword information, allowing for a more nuanced understanding of word structure, particularly beneficial for languages with rich morphology, and it effectively addresses the out-of-vocabulary problem by generating embeddings for words not seen during training.

This approach also aligns with more advanced neural architectures such as Convolutional Neural Networks [Pham et al., 2016], Recurrent neural networks [Mikolov et al., 2011], and Long Short-Term Memory Networks [Verwimp et al.,]. These architectures have been leveraged not only for context-based word prediction but also for directly addressing specific NLP challenges such as named entity recognition and chunking [Peters et al., 2017], as well as machine translation [McCann et al., 2017], bypassing the intermediate step of word prediction.

Parallel to these developments, Global Vectors for Word Representation (GloVe) were introduced in [Pennington et al., 2014], fundamentally differing from other embeddings in their construction. Initially, a co-occurrence matrix

X is constructed, where each element x_{ij} represents the frequency of word i occurring alongside word j in the corpus. Subsequently, the probabilities of the occurrence of word i in the context of word j are estimated. The final model is defined by the following function:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$

where w_i is the i-th word vector, w_j is the context-free j-th word vector, and b_i and b_j are bias terms for i-th and j-th words respectively. The function $f(X_{ij})$ is a weighting function applied to each pair of words, which helps to adjust the influence of rare and frequent co-occurrences differently. The function J seeks to minimize the squared difference between the predicted value and the actual logarithm of the words co-occurrence probability. This adjustment ensures that words appearing in similar contexts have similar vector representations. While GloVe successfully integrates both local and global statistical properties of the corpus, providing a robust method for learning word representations, these embeddings are static and do not account for word order, rendering them less effective compared to more advanced models.

Embeddings from Language Models (ELMo), introduced in [Peters et al., 2018], represents one of the first implementations of deep contextual embeddings. Unlike static word embeddings, which assign a single, context-independent representation to each word, deep contextual embeddings are aware of the surrounding text within a sentence. This means that a word like "host" would have different embeddings in "host plant" versus "host family", reflecting its different usages. ELMo achieves this contextual sensitivity through leveraging a deep bidirectional LSTM network, outlined in (see Section 2.1.1). This network aims to predict the next word based on its preceding context and the previous word based on its following context. Initially, ELMo constructs a character-based embedding for each word, and then processes these embeddings with the LSTM to account for the context in which each word appears. This method results in rich word representations that effectively capture both syntax and semantics, which are especially useful for tasks such as named entity recognition and part-of-speech tagging, which require a nuanced understanding of these linguistic aspects [Collobert et al., 2011, Pilehvar and Camacho-Collados, 2021]. The success of ELMo highlights their ability to provide distinct vector representations for the same word in different contexts, reflecting its varied meanings.

The introduction of attention mechanisms [Vaswani et al., 2017] has further revolutionized Natural Language Models by enabling the model to focus on different parts of the input sequence when predicting a word, thereby capturing even more complex dependencies. The first and classic

LM which leverages the transformer architecture is Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2019]. It is based on self-attention mechanisms which allows to weigh the importance of different words within a sentence. Unlike previous models that processed text in a single direction (either left-to-right or right-to-left), BERT considers the full context of a word by reading the text bidirectionally. Although ELMo embeddings consider both preceding and subsequent context, it processes one direction at a time during training. In contrast, BERT simultaneously considers both directions. This is a fundamental feature enabled by the Masked Language Modeling task (see Section 2.2.1), which helps in understanding the meaning of ambiguous words based on the context from both sides. In the MLM task, 15% of the tokens in each sequence are replaced with a special mask token, requiring the model to predict the masked words. This innovative approach makes BERT significantly more efficient at capturing long-range dependencies and handling complex linguistic patterns compared to RNNs and LSTMs, setting the groundwork for many subsequent advancements in the field of Natural Language Processing.

BERT has numerous variations, each tailored for specific enhancements or efficiencies. For instance, RoBERTa (Robustly optimized BERT) [Liu et al., 2019b] is an optimized version of BERT by refining the training process. One of the key modifications in RoBERTa is the introduction of dynamic masking, where tokens are masked anew before each training epoch, as opposed to being masked just once before all training starts. XLM (Cross-lingual LM) [Lample and Conneau, 2019] extends BERT's methodology to accommodate multiple languages. ERNIE (Enhanced Language Representation with Informative Entities) [Liu et al., 2023a], integrates knowledge graphs to enrich its language understanding. DistilBERT [Sanh, 2019] offers a lighter model that retains most of BERT's effectiveness but with fewer parameters and faster training. Additionally, ELECTRA [Clark et al., 2019b] introduces a novel training approach by using corrupted plausible tokens instead of masked ones, enhancing learning efficiency and model performance. The ongoing changes to the BERT framework show its versatility and adaptability, as researchers explore different aspects of the model to improve its effectiveness in various language processing tasks. These adjustments address specific challenges such as computational efficiency, multilingual support, and incorporation of external knowledge, enhancing the framework's usefulness in a variety of applications. Therefore, using BERT-based models in our research helps ensure comparability with previous and future studies, given that BERT has become a widely accepted standard in the field.

The ongoing evolution of the BERT framework, with its focus on addressing specific challenges and enhancing its versatility in various

language processing tasks, aligns with the broader landscape of transformer-based language models. While BERT emphasizes bidirectional understanding and adaptability across tasks, the Generative Pre-trained Transformer (GPT) models, first introduced in [Radford et al., 2018], represent another family of language models that leverage transformer architecture. Unlike BERT, which is bidirectional, GPT focuses on CLM pre-training (see Section 2.2.1), attempting to predict the next token based on all preceding tokens. The model is subsequently fine-tuned for specific tasks such as text classification, summarization, and question answering. One of the primary advantages of GPT models is their robust performance in both few-shot and zero-shot scenarios, even without task-specific pre-training [Brown et al., 2020a]. However, a notable drawback of these models is their large size and significant computational demands, which restrict their practical deployment in production environments. The GPT model has undergone several improvements over time. GPT-2 [Radford et al., 2019] and GPT-3 [Brown et al., 2020b] were trained on a more extensive datasets, and significantly increased the number of parameters compared to their predecessors. A notable advancement in GPT-3.5, also known as Instruct-GPT, [Ouyang et al., 2022] is its training protocol that incorporates human feedback, which refines its ability to follow instructions and generate more relevant outputs. The latest iteration, GPT-4 [Achiam et al., 2023], is the most advanced and efficient among GPT models, though specific technical details that explain its superior performance have not yet been publicly disclosed.

The T5, or Text-to-Text Transfer Transformer, introduced in [Raffel et al., 2020], is a model pre-trained to reconstruct a missing sequences of consecutive tokens. This text-to-text framework implies that T5 processes input text through an encoder, and generates output text using a decoder. The primary distinction between BERT and T5 lies in how they handle input and output transformations. BERT focuses on restoring masked tokens within a sentence, leaving the unmasked tokens unchanged. T5, on the other hand, adopts a more comprehensive approach by generating the missing elements of the text directly, without retaining the original input format.

For instance, given the input sentence "I enjoyed watching the new [MASK] movie with my friends", BERT would directly predict the masked word to complete the sentence, such as "I enjoyed watching the new horror movie with my friends". Conversely, T5 reformulates this task by potentially taking an input like "predict missing: I enjoyed watching the new [blank] movie with my friends" and producing the output "horror", focusing solely on the missing word without reproducing the entire sentence.

This flexibility allows T5 to dynamically generate textual responses, making it superior for tasks requiring more complex text generation.

However, the benefits of T5 come with trade-offs, including longer training times, larger model sizes, and challenges in model interpretability due to its extensive capabilities.

Bidirectional Auto-Regressive Transformer (BART) [Lewis et al., 2020] is a Sequence-to-sequence Transformer that is pre-trained to reconstruct corrupted noisy text. More precisely, it is composed of an encoder as in BERT and a decoder as in GPT. Unlike models that only replace masked tokens, BART is capable of restoring entire sequences where tokens have been masked, deleted, or permuted. The pretraining process consists in randomly reordering the original sentences and using a new masking approach where a sequence of tokens is replaced with a single mask token. While designed for text generation, BART can also effectively handle comprehension tasks.

Generally, these models perform exceptionally well when used within the scope of their design, specifically for the tasks and domains on which they were trained. However, their effectiveness diminishes when applied to text from highly specialized domains, resulting in poor output quality. This decline in performance can be attributed to several factors (for more details see Section 2.4.1):

- **Domain specific vocabulary**: Specialized fields often use terms that general models are not trained to understand
- **Varied text sources**: Texts from different origins may contain unique syntactic structures and stylistic elements that are unfamiliar to the model.
- **Fast evolution of knowledge**: Specialized domains frequently evolve, introducing new concepts and terminology that models trained on static datasets cannot consider.

Strategies to address these limitations are discussed in Section 2.3.5.

This section has covered a variety of foundational models in neural language modeling, from the pioneering approaches like Word2Vec and GloVe to more sophisticated architectures such as ELMo, BERT, GPT, and T5. Each model has contributed uniquely to the field, providing deeper insights and tools for handling the complexities of language through neural networks. While this discussion is not exhaustive of all the models developed, it includes those that have been particularly influential or represent significant technological steps forward. Subsequent sections will explore specific pre-trained models and advancements that build on these foundational techniques.

2.2.5 . Domain-specific Language Models

The training of language models often relies on text data from general domain, which encompasses a broad range of topics typical in everyday

language use. However, models trained on texts from this general domain might not fully capture the nuances of specialized domains like biomedical or financial literature. When dealing with such domains, domain-specific models become essential, because they leverage domain-specific knowledge and can achieve superior performance. There are many various domain-specific LMs. While there exist numerous domain-specific language models, this section will provide an overview of the models relevant to the present research.

BioBERT, as introduced in [Lee et al., 2020], stands out as a specialized version of BERT fine-tuned explicitly for biomedical NLP. Its training on PubMed [White, 2020] abstracts and PubMed Central [Roberts, 2001] full-text articles ensures its optimization for tasks crucial to the biomedical field, such as named entity recognition, relation extraction, and question answering. Given our focus on the biological domain, BioBERT emerges as a highly relevant tool for our research.

SciBERT [Beltagy et al., 2019] is pretrained on scientific literature, including papers from the computer science, physics, and other scientific domains. While not specific to plant epidemiology, it can be adapted for tasks in the biological and epidemiological domains. Evidence supports its successful application in biomedical event extraction tasks [Mulya and Khodra, 2023]. However, recent studies suggest that its performance might not be optimal for agricultural domain-specific tasks [D'Souza, 2024].

ChouBERT [Jiang et al., 2022a] is a pre-trained language model designed to extract knowledge from French plant health bulletins and identify tweets related to agricultural contexts. It demonstrates an improved performance over the CamemBERT model, a variant of BERT trained on French language texts. However, given our study's focus on English language analysis, we avoid using it.

While the models mentioned above are highly relevant to our research, there are numerous other domain-specific language models tailored to various fields, each contributing uniquely to the advancement of knowledge in their respective areas. It is important to note, however, the absence of a specialized language model for the plant health domain in English, which underscores a significant opportunity for future development.

Language Models are often adjusted and applied to a particular task, designed to generate, process or understand textual data. While the previous section highlighted the adaptation of language models to specialized domains, the following section will explore a task this thesis focuses on: Named Entity Recognition. Specifically, we will define the task, outline its methodologies, discuss its challenges, and examine its applications in various fields.

2.3 . Named Entity Recognition

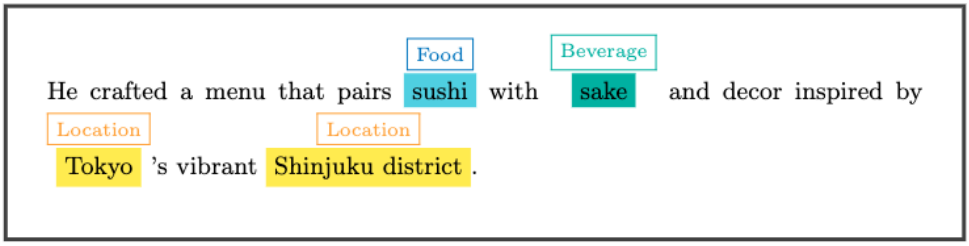

Figure 2.10: **Named Entities Recognition task.** This example illustrates labeled named entities in a textual passage.

2.3.1 . Task Definition

One of the fundamental tasks of Natural Language Processing is Information Extraction, which aims to extract structured information from unstructured text [Chen et al., 2022]. This process encompasses several subtasks, including Named Entity Recognition, Relationship Extraction, Event Extraction, Coreference Resolution, Normalization, Sentiment Analysis, and others. Named Entity Recognition (NER) is a key component of Information Extraction that focuses on identifying and classifying named entities present within a text into predefined categories (see Figure 2.10). These categories typically include, but are not limited to, names of persons, organizations, locations, dates, diseases or numerical values such as monetary amounts and percentages. NER primarily focuses on nominal and adjectival groups, with verbal groups considered only in specific conditions, particularly for actionable events such as "launched a product", "discovered a new species", or "arrested a suspect".

The primary goal of NER is to analyze unstructured text and extract structured and meaningful information. This process involves two main steps: detection and classification. Detection requires identifying the boundaries of named entities within the text—essentially, determining where an entity begins and ends. Classification involves assigning each detected entity to a specific category based on its context and characteristics. However, it's important to note that in practice, most of NER methods integrate these tasks into a single step by predicting a label for each token directly.

While some entities like proper nouns (e.g., "New York City") or collocations (e.g., "Heart Disease") are straightforward to recognize due to their fixed and consistent nature, the complexity of NER lies in the linguistic variability and ambiguity inherent in natural language. For example, the same word or phrase can function as a named entity or common noun depending on the context, e.g., "Java" as an island in Indonesia versus "Java" as a programming language. Additionally, terms often have synonyms that could be abbreviations (like "HBP" for "High Blood Pressure") or metaphorical

names (such as "Olive Tree Killer" for "Xylella Fastidiosa"), which are very complicated to identify. The system should also be able to recognize concepts it was never adjusted to, such as recognizing "NYC" as a location, even though it was adjusted for "New York City" only.

Furthermore, NER systems often face the challenge of handling *nested entities*, where one named entity is embedded within another, such as the location entity "America" within the organization entity "Bank of America". These scenarios require precise models that can discern multiple layers of entity categorization within a single context. Another significant challenge is *discontinuous entities*, which consist of parts that are separated from each other by other words. For example, the phrase "countries of South Asia" in "countries of Europe and South Asia" constitutes a discontinuous entity labeled as "location", referring to a specific region.

Overcoming these challenges requires sophisticated linguistic models and algorithms capable of understanding context and distinguishing between different uses of language.

2.3.2 . Evaluation

A standard evaluation measure for NER tasks is the F-measure, or F1-score, which combines precision and recall into a single performance indicator. Precision assesses the accuracy of the named entities correctly identified by the model out of all entities it identified, reflecting the model's exactness. Conversely, recall measures the proportion of correctly identified entities out of all the entities the model should have detected, thus showing the model's completeness.

A model with high precision but low recall indicates that while the entities it identifies are likely correct, it misses many relevant entities. This scenario may be preferable in situations where the accuracy of each identification is critical, such as in security-related contexts—for instance, monitoring communications for potential threats or screening individuals at checkpoints—where high precision ensures that the resources are concentrated on genuine threats, minimizing false alarms.

On the other hand, low precision with high recall suggests the model identifies most entities but also incorrectly labels non-entity text as entities. Prioritizing recall over precision might be more appropriate in scenarios where capturing as much relevant information as possible is paramount, such as during emergencies like natural disasters or public health crises. This approach is particularly useful if human review of the results, such as for epidemiological alerts, is planned. Here, systems prioritizing high recall could quickly extract all potentially relevant information from various texts, accepting the risk of including some irrelevant data to ensure comprehensive coverage.

The F-measure is the harmonic mean of precision and recall and is given by the following formula:

$$F1 = 2 \cdot \left(\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \right)$$

This formula ensures that both precision and recall are balanced, making the F1-score a crucial metric in scenarios where it is equally important to avoid false positives (incorrectly labeled entities) and false negatives (missed entities). This balanced approach is essential for developing effective NER systems that are adaptable to various practical scenarios, ensuring both reliability and robustness. Additionally, when evaluating the F1-score across multiple classes or datasets, averaging methods such as micro and macro averaging come into play. Micro averaging calculates global totals for true positives, false positives, and false negatives across all classes before computing the F1-score, making it sensitive to class imbalance. In contrast, macro averaging computes the F1-score separately for each class and then averages these scores, considering all classes are equal irrespective of their frequency.

2.3.3 . Datasets

A dataset is a systematically organized set of data collected or extracted from various sources, intended for analysis to draw conclusions, test hypotheses, or develop predictive models. In the realm of NER, a dataset is a collection of texts where each word is labeled with a specific entity type label or marked as not belonging to any entity type. Below are summaries of commonly used datasets for NER along with some domain specific datasets. It is important to note that while these datasets are representative, they are only a small fraction of the many available for NER research. For a more comprehensive list of NER datasets, researchers may refer to platforms like Papers with Code[1], which catalogues a wide array of datasets across various tasks and domains.

The Named Entity Recognition task was introduced by [Grishman and Sundheim, 1996] during the Message Understanding Conference (MUC). The goal was "to identify component technologies from the field of information extraction that are practical, largely domain-independent, and capable of being automated with high accuracy". Alongside this task, the MUC corpus was also introduced, which initially included only locations, organizations, and persons as named entities. However, this dataset is no longer in use as it became outdated and limited in scope, failing to meet the evolving needs of NER systems and the complexities of modern language data.

[1] https://paperswithcode.com/datasets

The most commonly used dataset for NER is CoNLL-2003 [Sang and De Meulder, 2003]. It consists of texts in both German and English, comprising named entities across four distinct categories: individuals, organizations, locations, and miscellaneous entities that do not align with the first three types. Primarily composed of news articles, the corpus showcases a vocabulary representative of the general domain. Since it is a widespread dataset, its usage ensures that our results derived from it are comparable to current standards and research outcomes in the field.

OntoNotes dataset [Weischedel et al., 2013] was developed to annotate a large textual corpus in Arabic, English, and Chinese languages, composed of diverse sources such as weblogs, broadcast talk shows, telephone conversations, and news articles. The dataset's annotations extend beyond Named Entity Recognition to encompass structural elements such as syntax and predicate-argument structures, as well as shallow semantic information, providing a rich resource for multifaceted linguistic analysis. However, recent findings have highlighted annotation errors within the corpus[Bernier-Colborne and Vajjala, 2024], leading to our decision not to use it.

WNUT 2016 [Strauss et al., 2016] and WNUT 2017 [Derczynski et al., 2017], both originating from Twitter data [Ritter et al., 2011, Baldwin et al., 2015], respectively include ten and six Named Entity types, covering a wide range of general domain topics, such as company, facility, movie, person, product, location. This collection spans a broad spectrum of topics within the general domain, including shooting events and cybersecurity incidents. Unlike its predecessor, WNUT 2017 includes comments exceeding 140 characters to capture a variety of writing styles and unique characteristics. While it retains the Twitter texts used in WNUT 2016, it expands its scope by incorporating additional comments from Reddit[2], Stack Exchange[3], and YouTube[4]. This expansion aims to cover a wider array of subjects, from geographical nuances to specific topic-related events, thereby facilitating a comprehensive approach to data mining. However, we did not use this corpus in our experiments, as we aimed to test our approach on datasets from different domains with minimal overlap in entity types, and the WNUT entities largely overlap with those in the CoNLL dataset we had already selected for our experiments.

The Wikigold corpus [Balasuriya et al., 2009] consists of Wikipedia articles across nine languages, each manually annotated with named entities such as individuals, organizations, and locations. For more granular analysis, these main categories are further divided into subcategories such as Geopolitical

[2] https://www.reddit.com/?rdt=34882
[3] https://stackexchange.com/
[4] https://www.youtube.com/

Entities, facilities, and towns within the location entity type, providing a detailed framework for entity recognition across diverse linguistic contexts. Although this corpus is relevant to our study, we did not use it for the same reasons as the WNUT dataset.

The MIT Movies and Restaurant corpora [Liu et al., 2013a] delve into specific domains, offering rich datasets for analyzing movie-related information and restaurant-related conversations through short reviews, descriptions, and queries. The MIT Movies corpus covers 19 Entity types including actors, awards, genres, soundtracks, among others, offering a rich dataset for analyzing movie-related information. In parallel, the MIT Restaurant corpus contains 8 annotated Entity types, including dishes, cuisines, amenities, locations, hours, ratings, among others, providing a comprehensive dataset for exploring restaurant-related conversations. Both corpora are domain and format specific because of user-generated content and a distinct set of entities, making them particularly suitable for our research.

The NCBI Disease Corpus, developed by [Doğan et al., 2014], is centered on human diseases and comprises 793 PubMed[5] abstracts. It contains mentions of four distinct entity types: Composite Mentions, Modifiers, Disease Class Mentions, and Specific Diseases. Furthermore, this corpus enriches Named Entity Recognition with Concept Normalization annotations, using MeSH [Lipscomb, 2000] and OMIM [Hamosh et al., 2000] codes. Thus, with its scientific text style and specific entity set, the NCBI Disease Corpus is a relevant resource for our research.

Bacteria Biotope [Bossy et al., 2019] is a corpus specifically focused on the identification and annotation of bacteria and their phenotypes and geographical locations. It includes three labeled entity types: microorganism, habitat and phenotype. Additionally, this corpus is enriched with annotations for concept normalization and relation extraction, enhancing its utility for complex analysis tasks. Therefore, Bacteria Biotope, being a corpus specific to the biological domain, is highly relevant to our study.

This overview of datasets for Named Entity Recognition illustrates the variety and depth of resources available in the field. The diversity in the types of named entities underscores the complexity of the task. Each dataset, whether general or domain-specific, serves as a crucial tool for developing and evaluating NER models, catering to a broad spectrum of linguistic challenges. Consequently, the selection of a dataset is essential for designing and training effective NER models as it directly impacts the breadth and specificity of recognizable entities by the model.

2.3.4 . Annotation Standards

[5] https://pubmed.ncbi.nlm.nih.gov/

The process of labeling or annotating named entities in text is pivotal. It lays the groundwork for training machine learning models accurately. However, this task is not without its complexities, mainly due to the diversity of annotation standards, guidelines and formats employed across different datasets and projects. These standards dictate how text should be annotated, including the definition of entity types, the format of the annotation, and how to deal with ambiguities or entities that span multiple words. Some of the widely recognized annotation standards in the NER domain include:

IOB (Inside, Outside, Beginning)

One of the simplest and most fundamental tagging schemes of NER is IOB, that classifies tokens as Inside an entity, Outside any entity, or at the Beginning of an entity. Each line in an annotation file typically pairs a token (often a word) with a tag that combines its position (I, O, or B) and entity type. While effective for basic NER tasks, IOB can struggle with nested and discontinuous entities. For example, in the phrase "Bank of America", "Bank" would be annotated with the label "B-org", and "of" with "I-org". However, it becomes challenging to annotate "America" as it is part of an organization entity and simultaneously represents a separate location entity. This ambiguity highlights the limitations of the IOB scheme in dealing with complex entity structures that might be better served by more detailed schemes like BIOES.

BIOES (Beginning, Inside, Outside, End, Single)

BIOES, also known as BILUO (Beginning, Inside, Last/End, Unit/Single, Outside), extends the IOB tagging scheme by incorporating two additional tags. These additional tags explicitly mark the End of an entity and identify Single-token entities.This additional granularity helps models more accurately predict entity boundaries. However, it still does not fully accommodate the annotation of nested and discontinuous entity structure.

BRAT stand-off format

BRAT (Brat Rapid Annotation Tool) is a web-based tool for text annotation that also defines its own annotation format. Annotations are stored separately from the text in dedicated files, a format known as stand-off annotation, which is particularly useful for avoiding copyright issues. Each line in the annotation file corresponds to a named entity and includes an annotation ID, an entity type, indices of the character string in the text, and the corresponding character string itself. For example, in the phrase "Harvard University located in Cambridge" the annotation file would contain two lines: "T1 Organisation

0 18 Harvard University" and "T2 Location 30 39 Cambridge". This structure allows to consider nested and discontinuous entities, providing flexibility for annotating complex linguistic structures that may overlap or be separated by intervening text. Additionally, BRAT supports normalization annotations, which serve as the subsequent stage in the information extraction process following NER.

PubTator format

Like BRAT, PubTator is an annotation tool that specifies its own format. Developed by the National Center for Biotechnology Information (NCBI), it is designed to meet the specific needs of the biomedical community. In PubTator, both annotations and the associated texts are contained within the same file. Each annotation entry includes a text ID, the corresponding character string, its indices in the text, an entity type, and an identifier linking to an appropriate database or ontology.

BioC format

Developed as part of the BioCreative (Biological Text Mining and its Applications in Biomedicine) community initiative, BioC is an XML-based format designed to facilitate the interoperability of biomedical text mining systems. In this structured format, each annotation is linked to the corresponding text through the hierarchical organization of an XML tree. Annotations in the BioC format include comprehensive details such as the entity type, the exact character sequence from the text, the start index, the length of the annotation, and a unique identifier from a relevant database or ontology. This structure not only ensures precision in text annotation but also promotes consistency and ease of data integration across different text mining and processing tools. However, it does not naturally accommodate discontinuous entities, which can be crucial for capturing complex biological phenomena.

Silver Annotation

Unlike the manually curated methods discussed above, silver annotations are automatically generated by one or several methods, often averaged or weighted by confidence factors. Although commonly used in semi-supervised learning environments and evaluated in few-shot scenarios due to their minimal reliance on labeled data, silver annotations must be used cautiously, because the quality of labeled data has an impact on the performance of machine learning algorithms. Therefore, while they provide a scalable

annotation method, their reliability may not always match that of manually curated datasets.

There are many more annotation formats used for NER, but only the most classic and widely used are listed here. This diversity of formats can be both an advantage and a disadvantage. On one hand, it allows users to select the format that best suits their specific needs; on the other hand, it necessitates the reformatting of annotations when applying systems across different corpora. The choice of annotation standard can significantly impact the training and performance of NER models, given the consideration - or lack thereof - of nested and discontinuous entities, emphasizing the importance of careful selection based on the unique objectives and challenges of each project. In our research, we used the BRAT format because it accommodates all the complex phenomena we needed to model, including discontinuous and nested entities.

2.3.5 . Techniques and Models Overview

Rule-based Approaches

Traditional approaches to NER were predominantly rule-based, employing expertly crafted sets of predefined rules. These rules, which hinge on linguistic, syntactic, and domain-specific knowledge, are meticulously tailored to specific domains to ensure accuracy and efficiency. Due to their domain-specific nature, rule-based systems are highly precise but lack generalization performance across different domains and require manual effort in rule creation, alongside advanced domain expertise.

For instance, regular expressions and dictionary based approaches were employed by [Packer et al., 2010, Etzioni et al., 2005, Sekine and Nobata, 2004, Zhang and Elhadad, 2013, Kim and Woodland, 2000, Hanisch et al., 2005] and compared with ML approaches. Surprisingly, regular expressions outperformed ML approaches in some cases. This performance can be attributed to the inclusion of specific mentions in the dictionaries on which regular expressions are based, and their absence in the training data used by ML algorithms.

[Eftimov et al., 2017] proposed a rule-based NER method called drNER for extracting dietary information, marking a novel contribution to the field. Their two-phase approach begins with entity detection and proceeds to entity extraction, validated across texts from scientific websites and research articles.

Regular expressions were also used to complete ML approaches. Thus, [Sari et al., 2010] combined rule-based extraction with semi-supervised learning to identify entities in accident records. Using tools like the Stanford Part-of-Speech tagger, they extracted patterns to classify entities into categories like *date* and *location*, assessing their method's accuracy

through Exact Match evaluation.

[Korkontzelos et al., 2015] introduced a voting mechanism to combine predictions from different ML models and developed string-similarity metrics based on common suffixes in drug names, using these metrics to generate regular expressions that enhance dictionary-based tagging accuracy.

The FoodIE approach, presented in [Popovski et al., 2019], is designed to extract food entities from unstructured textual data, including recipes and dietary recommendations. It leverages existing terminologies and syntactic patterns to identify and classify food-related terms without requiring an annotated corpus, addressing a significant gap in food entity research within biomedical literature. This system integrates text preprocessing, part-of-speech tagging, semantic tagging, and the recognition of food entities into a coherent workflow, highlighting its adaptability and potential for enhancing food-related information extraction, which is particularly beneficial for public health applications. However, since this approach relies heavily on a set of domain-specific rules, its adaptability to other types of data or domains may be limited.

Several other rule-based systems, such as those described by [Boulaknadel et al., 2014], [Elsayed and Elghazaly, 2015], [Pushpalatha and Thanamani, 2019] demonstrated the applicability of rule-based approaches in information extraction across various languages and domains. Despite their precision and reliance on detailed linguistic and syntactic knowledge, these systems are constrained by their domain-specific focus and require a substantial amount of human expertise to create effective rules. Furthermore, they cannot be easily adapted to new domains [Li et al., 2020]; in fact, it is often more feasible to develop a separate rule-based system for each new domain. For these reasons, machine learning approaches have become more common and widely used in the field.

Machine Learning Approaches

The state-of-the-art NER models rely on supervised Machine Learning algorithms such as BiLSTM+CRF [Wang et al., 2021], BERT [Devlin et al., 2019], generative models [Shen et al., 2023] or prompt engineering [Wang et al., 2023a].

Supervised methods are applied to data that comes with predefined labels, training machines on this labeled dataset to either make predictions or classify the data based on the given task. In the context of Named Entity Recognition systems, selecting the appropriate learning algorithm is pivotal. The traditional supervised learning algorithms for NER tasks include Hidden

Markov Models (HMM), Conditional Random Fields (CRF) [Li et al., 2009], and Support Vector Machines (SVM) [Ju et al., 2011].

HMM algorithms were once popular for resolving NER tasks. The first use of HMM was introduced by [Bikel et al., 1997] and showed 93% of F-score on MUC-6 dataset, thus demonstrating its efficacy. Various modifications were explored by researchers [Zhou and Su, 2002, Morwal et al., 2012, Morwal et al., 2012]. However, HMMs do not effectively capture long-range dependencies in text, they require careful feature engineering and are less efficient than deep learning models.

CRFs have been also extensively used for NER [McCallum and Li, 2003, Settles, 2004, Krishnan and Manning, 2006], but with more varied approaches than HMMs. For instance, to address challenges like noisy and limited data in social media NER, particularly in the context of concise and sparse tweets, one of the early Machine Learning methods was introduced by [Li et al., 2009]. It capitalizes on tweet redundancy through a two-phase NER process. Initially, each tweet undergoes a preliminary labeling using a sequential labeler based on a CRF model (see Section 2.1.1). Subsequently, tweets sharing similar content are aggregated into clusters. Within these clusters, tweet labels are refined through a CRF model that assimilates cluster-level information, including the labels of the current word and its adjacent words across all tweets in the cluster.

Some researchers still use CRF in recent works. A statistical NER system for the Marathi language, developed by [Patil et al., 2020] uses a Conditional Random Field for identifying and categorizing named entities. Given Marathi's morphological richness, this approach effectively locates and categorizes named entities within the language, showing promising results in accuracy. However, incorporating more contextual knowledge could potentially enhance these results. However, this research is exceptional, as CRF models are rarely used alone nowadays. Instead, they are widely implemented on top of neural networks for enhanced performance.

SVM were also among the first models to achieve significant success in classifying named entities. Firstly applied in [Takeuchi and Collier, 2002], it outperformed HMM model on MUC-6 dataset. It was used by many researchers due to its efficiency on small datasets.

These models were considered to be highly effective in NER task before the advent of more advanced ML models, primarily due to their ability to model the sequential dependencies in text data. The advent of neural networks has marked a significant evolution in NER methodologies. One of the most widely used architectures, first applied in [Limsopatham and Collier, 2016], is a bidirectional LSTM model that automatically generates orthographic features from Twitter text, eliminating the reliance on manually crafted features. This approach improves the

model's capability to categorize entity mentions within the noisy and informal language typical of tweets. The model employs both character and word embeddings; it uses a convolutional neural network to create character-based word representations while also incorporating pre-trained word embeddings to capture deeper semantic and syntactic context. This model significantly outperformed previous methods on social media texts.

Researchers are not only refining existing methodologies but also embracing advanced techniques that leverage the strengths of both approaches. The integration of diverse algorithms, such as CRF and SVM, has demonstrated substantial improvements in dealing with complex data characteristics. For instance, methods like the two-phase NER process for social media and the dual-phase strategy in biomedical contexts illustrate the adaptation of these models to specific challenges such as sparse data and class imbalance. These developments pave the way for incorporating more sophisticated neural network techniques that promise even greater enhancements in NER performance.

In the biomedical field, a BERT-HMM-based NER system, introduced by [Li et al., 2021], was specifically designed to address the challenge of noisy labels from multiple sources. This system uses an HMM model to refine labels predicted by BERT, demonstrating how simpler methods can enhance the effectiveness of modern approaches when combined.

Among the most significant of the advancements in the field of model integration is the combination of Long Short-Term Memory (LSTM) (see Section 2.1.1) networks with Conditional Random Fields (CRF) (see Section 2.1.1), including their Bidirectional variants. The integration of BiLSTM with BiCRF, first applied to NER by [Panchendrarajan and Amaresan, 2018], effectively models label dependencies by considering both preceding and subsequent labels within the sequence. The architecture also incorporates pre-trained word embeddings, Part-of-Speech tags, and casing features. Demonstrating robust performance on the CoNLL-2003 dataset, it notably outperforms models using unidirectional CRF. Furthermore, it excels in identifying complex named entities and enhancing the detection of Miscellaneous entities due to its bidirectional capabilities.

Further advancements have been achieved with the introduction of BERT [Devlin et al., 2019]. BERT transforms the landscape of NER tasks by using a mechanism known as the transformer (see Chapter 2.2), which processes words in relation to all other words in a sentence, rather than one at a time. This allows BERT to capture the full context of a word by looking at both its left and right surroundings—a feature called bidirectional training. When applied to NER, BERT has shown remarkable success, outperforming previous models by a significant margin because it deeply understands the semantic relationships within text.

The WikiNEuRal approach [Tedeschi et al., 2021] combines two previously discussed approaches and uses BERT+Bi-LSTM+CRF. This method harnesses the capabilities of pretrained language models within a framework that incorporates knowledge-based techniques and neural models. By leveraging the hyperlinked structure of Wikipedia for data extraction and annotation, WikiNEuRal distinguishes named entities from general concepts (e.g. *appartment*, *plane*, etc.) in Wikipedia articles. It also employs a validation process for silver annotations and identifies previously unlabeled entities within this silver data, thereby overcoming the limitations of traditional heuristic methods. The approach also incorporates domain adaptation algorithms, enhancing its performance across various testing settings and demonstrating significant improvements in span-based F1-score points on standard benchmarks.

The NER task can also be solved using a generative approach, such as SC-NER model [Wang et al., 2019]. This model initially employs a classifier to determine the presence of entities in sentences, followed by a sequence-to-sequence (seq2seq) framework that incorporates both an encoder and a decoder based on Long Short-Term Memory networks. The output is a sequence of labels instead of words (e.g. for "Albert Einstein was born in Ulm" a sequence can be "B-per I-per O O O B-loc"). Preliminary results demonstrate that SC-NER outperforms classic models on a custom dataset of patent documents in precision and recall, particularly in recognizing specific entity types such as materials, highlighting its potential suitability for specialized domains.

DiffusionNER [Shen et al., 2023] redefines a NER task as a denoising diffusion process to refine the boundaries of named entities from noisy spans. Generally, a denoising diffusion process creates images or generates data by starting with a completely random pattern and gradually removing the randomness (or "noise") to form a structured, meaningful image. In the context of DiffusionNER, this process begins with an approximate initial guess of entity locations in the text and methodically refines these guesses by progressively eliminating errors and uncertainties. This method not only enhances the accuracy of identifying entity boundaries but also enables the model to handle an arbitrary number of entities dynamically during evaluation, a significant advantage in practical scenarios.

The W^2NER method [Li et al., 2022b] uses a combination of BERT and BiLSTM for generating contextual word representations, and applies convolutions for refining these representations. Relations between words are categorized into Next-Neighboring-Word and Tail-Head-Word-* relations, aiding in the identification of entity boundaries and types.

The Universal NER approach [Zhou et al., 2023] involves refining the ChatGPT model using a technique called distillation, which involves a smaller,

less powerful model learning to imitate the predictions of a larger, fully trained model. Despite its smaller size, UniversalNER outperforms ChatGPT and other BERT-based models in recognizing named entities across widely used datasets like ConLL-2003, MIT Movies, NCBI diseases, WikiNeural, and others in both fine-tuned and zero-shot settings. However, this method depends on the availability of a large pre-trained model, making it less versatile for situations where such models are not accessible. Distillation primarily serves as an optimization strategy to enhance performance and efficiency, rather than a domain adaptation technique.

Thus, supervised learning techniques currently represent the state-of-the-art in Named Entity Recognition systems. As technology advances, the integration of classical algorithms like CRF and SVM with advanced neural network architectures such as BERT has transformed the field of NER. However, these methods rely on a substantial amount of labeled data [Jehangir et al., 2023]. The performance of these supervised methods typically improves with an increase in training examples, often reaching state-of-the-art results when sufficient data is available. Yet, in practical scenarios, especially in low-resource languages or specialized domains, acquiring a large volume of annotated data is either costly or impractical. This limitation highlights the necessity for exploring unsupervised methods, which do not depend on labeled data and exploit inherent patterns within the dataset itself.

Unsupervised methods Unsupervised learning techniques are applied to unlabeled data, to raw texts in NLP, although labels may be used for evaluating the models. Two primary strategies in unsupervised learning include association and clustering. The association technique focuses on detecting patterns and relationships among variables by analyzing their joint occurrences. Conversely, clustering involves organizing objects into groups (clusters) where the members of each group are more alike to each other than those in different groups, which helps reveal natural classifications within the data. These methods are crucial in contexts where labeled data is scarce or costly to obtain.

In the realm of biomedical NER, the approach outlined by [Zhang and Elhadad, 2013] stands out for its flexibility across different semantic categories and textual formats. This method, which processes raw texts, is based on leveraging terminologies, shallow syntactic patterns, and statistical data from the corpus. Entity types are modeled through classes of a knowledge base, which also helps in normalizing concepts. While their technique demonstrated promising results on clinical notes and biomedical datasets, it does not take into consideration nested and discontinuous entities, which are prevalent in complex biomedical texts.

A cross-domain unsupervised domain adaptation model was proposed by [Peng et al., 2021]. It uses adversarial training of a classic model coupled with entity-aware attention to reduce domain distribution shifts. The adversarial training component aims to minimize the misalignment of entity features during the learning process. However, the model's performance is likely highly dependent on the quality and representativeness of the source domain data. If the source data does not adequately capture the diversity of entity types or contextual usage found in the target domain, the model's effectiveness could be compromised.

The Knowledge Augmented Language Model (KALM) [Liu et al., 2019a] integrates a RNN-based neural language model with information about entities types, which are organized into groups within the knowledge base. KALM improves language modeling by using a gating mechanism that decides whether a word is treated as a general vocabulary item or as an entity, enhancing the model's ability to handle named entities and generalizing across entity classes. This approach enables KALM to recognize named entities in an unsupervised manner, demonstrating the potential of predictive learning combined with entity knowledge to enhance the training of deep learning models. The model significantly reduces perplexities in language modeling tasks and achieves competitive accuracy in NER. Although effective, the model requires access to a knowledge base, which may be difficult to obtain for certain domains.

CycleNER, introduced in [Iovine et al., 2022], is a generative approach that uses unannotated sentences along with a set of named entity samples which reflect the same entity distribution found in the target texts. This method consists of two core components: Sentence-to-Entity and Entity-to-Sentence, both implemented as sequence-to-sequence networks. These components are trained through a bi-directional cycle, allowing the system to convert sentences to entities and *vice-versa*, thereby leveraging the cycle-consistency concept. CycleNER was evaluated on several benchmarks, showing competitive results close to state-of-the-art supervised methods.

Overall, the use of unlabeled data by unsupervised methods makes the training process cost-effective compared to traditional supervised methods. Despite the advancements brought by these unsupervised methods, they generally fall short of the accuracy achieved by supervised approaches.This gap has led to the further exploration into alternative strategies that aim to bridge the robustness of supervised learning with the scalability of unsupervised techniques.

Semi-supervised methods Semi-supervised learning approaches for NER offer promising solutions by leveraging both labeled and unlabeled data, aiming to address the scarcity of annotated corpora in specific domains.

These methods blend the strengths of supervised learning with the ability to harness large volumes of unlabeled text, enhancing model performance without extensive manual labeling efforts.

One notable approach is kNN-NER [Wang et al., 2022] that integrates k-nearest neighbor (kNN) retrieval with traditional NER models to enhance performance, particularly in managing long-tail cases without extensive training datasets. The proposed model retrieves similar examples from a cached training set during inference, thereby reducing the need for memorization and allowing the model to generalize better from less data.

A semi-supervised ensemble learning approach was introduced by [Ma et al., 2020]. It combines Conditional Random Field, Bidirectional Gated Recurrent Unit, and Bidirectional Long Short-Term Memory models. This approach implies an iterative training of the model with minimal labeled data alongside a substantial corpus of unlabeled text. By initiating training with a small labeled dataset for pre-training the base learners, which then collaboratively assign reliable labels to the unlabeled data through a tri-training algorithm.

Weakly supervised methods Weakly supervised learning shares similarities with semi-supervised learning. This distinction focuses on the type and quality of data each method employs to train models. While semi-supervised learning uses a combination of labeled and unlabeled data to enhance training, weakly supervised learning operates with labels that are noisy, inexact or incomplete [Zhou, 2018].

One of working weakly supervised methods to overcome data scarcity limit is distant supervision, which is based on the hypothesis that if specific words or phrases are labeled as entities in one corpus, they should similarly be labeled in other corpora, allowing for the generation of labeled data from unlabeled text. However, this approach can mistakenly label non-entity words as entities, leading to decreased performance. To address this issue, the Spy-PU algorithm [Zheng et al., 2021] treats these incorrectly labeled samples as unlabeled and introduce a semi-supervised method to reliably identify positive samples. This method strategically embeds a certain proportion of known positive samples into a set of unlabeled samples to help identify and confirm genuine positive samples within that unlabeled set. Demonstrating promising results, this method enhances the effectiveness of Chinese NER models across multiple public datasets, overcoming the limitations of sparse and inaccurately labeled training data.

The Bert-Assisted Open-Domain Named entity recognition with Distant Supervision framework [Liang et al., 2020] is another approach with distant supervision that uses a two-stage training process to effectively address the challenges of label scarcity and noisy data in NER. Initially, the RoBERTa model

is fine-tuned with distantly-matched labels. Subsequently, a teacher-student framework refines the training through the use of pseudo soft-labels, significantly enhancing the model's ability to handle incomplete annotations and improving overall quality.

Each of these works exemplifies the diverse strategies within semi-supervised and distant supervised NER, ranging from ensemble learning and unsupervised entity linking to original applications of distant supervision. Such approaches lay the groundwork for further exploration into how models can be fine-tuned for even more precise and contextually aware entity recognition. Nevertheless, there are other techniques that allows to improve NER systems particularly in scenarios where data is scarce. These will be discussed in the subsequent sections.

Prompt engineering Prompt engineering represents a distinct family of techniques in the landscape of machine learning, positioned as a method that focuses on modifying inputs to use pre-trained language models (PLMs) without the extensive need for new data labeling or model retraining. In the context of NER, it is aimed to optimize the recognition and classification of entities within texts, particularly in low-resource scenarios. By structuring input data through pre-designed prompts, researchers can leverage the intrinsic knowledge of PLMs without requiring extensive labeled datasets.

A prime example of this approach is PromptNER [Zhang et al., 2023a], a few-shot prompting approach. This model integrates a component for detecting the positions of entity spans and utilizes a classifier to determine entity types via prompts. Unlike traditional methods that rely on prototypical networks, PromptNER employs k-nearest neighbor search to leverage entity information from supporting examples effectively. This allows for direct fine-tuning on new support sets, facilitating a smoother transition from the training to the fine-tuning phase. However, PromptNER cannot recognize discontinuous entities, as it assigns only one position slot per entity, which might not capture multiple segments of a split entity. In addition, the number of entities that PromptNER can identify is limited by the number of position slots pre-determined during its training, which can lead to problems if an unexpected number of entities appears in the text.

A Prompt-based Text Entailment (PTE) approach, introduced by [Li et al., 2022a] redefines the task of Named Entity Recognition by framing it as a text entailment problem, which effectively leverages pre-trained language models in low-resource settings. In natural language processing, entailment involves evaluating whether a premise can logically imply the truth of a hypothesis. Using this framework, the PTE method generates specific prompts corresponding to each entity type and pairs these prompts with the original sentences. These combinations are fed into pre-trained models

that evaluate and score each potential entity type based on its congruence with the given sentence context. The entity type receiving the highest score is then selected as the correct classification. This approach streamlines the recognition process by focusing on individual words rather than larger text segments or n-grams, enhancing efficiency and adaptability in scenarios with sparse data. Nevertheless, the model's reliance on the binary outcome of a text entailment task (entailment *vs* non-entailment) may oversimplify the complexity and ambiguity inherent in human language, especially in handling entities that require contextual or background knowledge to be identified correctly.

The COntrastive learning with Prompt guiding for few-shot NER (COPNER) technique, proposed by [Huang et al., 2022]combines contrastive learning with prompts enriched with class-specific words. These words represent various entity categories, such as "individual" for "person", "place" for "location", and "company" for "organization". By embedding these prompts, the model leverages the semantic richness of Pre-trained Language Models. During training, COPNER aligns the word representations with these prompts. In the inference phase, these class-specific words serve as metric referents, enabling the model to classify new examples accurately. This method enhances previous prompt-based approaches by providing a more adaptive and versatile framework for entity recognition, particularly effective in specialized or complex areas where conventional models may struggle due to limited data.

QaNER, developed by [Liu et al., 2022], transforms the NER task into a Question Answering (QA) format. This method generates NER specific prompts for QA models. By fine-tuning QA models with a few annotated NER example, QaNER facilitates low-resource training and enables zero-shot learning capabilities. It outperforms other approaches in terms of computational efficiency and adaptability in low-resource and zero-shot scenarios on the MIT Movies [Liu et al., 2013a], MIT Restaurants [Liu et al., 2013a] and CoNLL-2003 [Sang and De Meulder, 2003] datasets.

While prompt engineering significantly enhances the capabilities of pre-trained language models in NER, it requires substantial computational resources and offers limited options in open-source tools. Moreover, the decision-making process in prompt engineering is often challenging to interpret because, unlike traditional learning strategies where features are explicitly designed and selected for model training, control over the model is indirect and mediated through prompts. Despite these challenges, whether employing supervised, unsupervised, semi-supervised, or prompt engineering, the effectiveness of the algorithm in a specific domain can be further improved through domain adaptation techniques, which will be explored in the following section.

Domain Adaptation techniques

Domain adaptation is usually seen as a specific form of transfer learning, a machine learning strategy where knowledge acquired from solving one (source) problem is applied to solve a related but distinct (target) problem. The principle behind this approach is that general insights gained from one domain can improve performance in another. Specifically, domain adaptation deals with the challenge of transferring knowledge between domains. A "domain" encompasses the data environment, including its features, distribution, and specific tasks. For instance, a model trained on English news articles might be adapted to analyze social media texts. Although both domains involve text and may share tasks like sentiment analysis or NER, their linguistic styles, usage, and content often differ markedly.

Traditional approaches to domain adaptation in NER frequently use difference between domains, as demonstrated in [Jia et al., 2019]. This method leverages both domain and task-specific embeddings to generate dynamic parameters for NER and MLM tasks. The core of this approach is the hypothesis that the overall behavior of the model is determined not by a unified set of parameters but by a combination of foundational settings (meta parameters) and the adjustments provided by task embeddings and domain embeddings. By decomposing the parameters into meta parameters and embeddings, the approach allows the model to flexibly adapt to various domains and tasks by mixing and matching these components, leading to more effective and versatile performance across different scenarios. This is achieved by training the model with a combination of both labeled NER data and unlabeled raw data for MLM task (see Section 2.2.1) from both source and target domains. One unique feature of this method is that it supports learning in new fields without needing specific examples from those areas, by automatically deriving target-domain NER parameters from source-domain data.

The study of [Peng et al., 2021] introduces an entity-aware adversarial domain adaptation network to bridge the domain gap for NER. The approach applies adversarial training to minimize the distributional differences in token probabilities across domains while paying special attention to entity features through an entity-aware attention mechanism. This dual strategy ensures that the model not only learns domain-invariant features but also prioritizes the alignment of entity-related characteristics, leading to superior performance in the target domain. This model demonstrates the efficacy of adversarial approaches in domain adaptation for NER by focusing on entity-level feature alignment.

In [Zheng et al., 2022] relationships between labels are modeled via graphs. By constructing label graphs in both source and target domains

and formulating the problem as a graph matching issue, this approach allows the transfer of label knowledge even when label sets do not overlap. Integrating label graphs with word embeddings enhances the model's ability to understand and use label-specific information, thereby improving cross-domain NER performance. This method highlights the potential of graph-based approaches in capturing and transferring complex label relationships across domains.

In [Zhang et al., 2021], domain adaptation is examined from a unique perspective by treating crowdsourcing for NER as its application. By considering crowdsourced annotations as domain-specific data, this approach implies applying cross-domain adaptation techniques to leverage the diversity in annotators' perspectives. An annotator-aware representation learning model is proposed to effectively capture domain annotator-specific features, demonstrating substantial improvements in NER performance with crowdsourced data.

The approach that captured our attention is described in [Ma et al., 2022]. In this method, the authors leverage two BERT models. The first BERT model encodes the input text along with its tokens in a normal way. The second model is dedicated to encoding the labels, specifically the entity types. Each label is transformed in a natural language form (e.g., "person" for "PER"), but conserves the IOB format (see Section 2.3.4), resulting in labels like "beginning person" for beginning of an entity "person" and "inside person" for continuation of an entity "person". These enriched label embeddings are then combined with text embeddings through a dot product operation. Predictions are then made by selecting the label with the highest dot product value. The model is trained on the Ontonotes dataset and evaluated on 5 datasets from diverse domains, such as biology, medical and news. This dual-encoder strategy enhances the model's ability to understand and categorize entities, particularly in few-shot learning scenarios where data is limited.

Our approach draws inspiration from the concepts outlined in [Ma et al., 2022]. Similar to the strategy described in that work, we use label semantics, transforming them through a BERT-based model. However, our approach does not entail model training for labels representation.

In summary, the explored studies present diverse strategies for domain adaptation in NER, from adversarial training and attention mechanisms to graph-based label transfers and innovative uses of crowdsourced data. Each approach addresses specific aspects of the domain adaptation challenge, whether it be reducing domain discrepancies, improving model robustness, transferring label knowledge across mismatched domains, or redefining the task. Collectively, these contributions enhance the adaptability and performance of NER models across varied and evolving domains. As we move to the specific challenges and strategies of NER in the context of plant health,

it becomes clear that such advancements in domain adaptation are critical in applying NER to more specialized and technically demanding fields.

2.4 . Named Entity Recognition and Domain Adaptation for Plant Health

In the field of plant health, quickly processing and analyzing textual information from real-time data sources like social media and news outlets is crucial for enhancing dynamic monitoring capabilities. This enables the development of alert systems that proactively inform farmers and agricultural professionals about new disease threats, allowing for early interventions to decrease risks and manage these outbreaks more effectively. These systems can analyze trends and patterns in the data, providing early warnings that are essential for maintaining plant health.

Furthermore, Named Entity Recognition supports decision-making processes by structuring detailed insights into plant health dynamics. This automation aids in the creation of extensive databases that document various aspects of plant health, including potential triggers, environmental factors, and effective treatment methods. Such databases offer organized, accessible knowledge to farmers, researchers, and agronomists, ensuring sustainable farming practices and optimal resource utilization.

Thus, Named Entity Recognition is a valuable tool in the plant health context, automatically extracting important information related to plant health. This includes the identification of a wide range of entities, from highly specific domain entities such as species (e.g. Xylella fastidiosa), symptoms (e.g., leaf scorch, wilting), pathogens (e.g.Flavescence dorée), diseases (e.g., Banana bunchy top disease), to generic entities that are common across various domains, such as locations (e.g., West Nile region) and dates (e.g., last month).

Applying NER in plant health involves understanding specific linguistic challenges present in agricultural texts. These challenges include accurately identifying relevant entities among common language ambiguity and scientific terminology complexity. It also involves processing information from diverse sources with varying levels of technical language. Such sources include academic research papers, technical reports, social media.

Furthermore, the significance of NER extends beyond text extraction; it plays a vital role in improving decision-making processes. Integrating NER into precision agriculture systems that combine data from multiple sources like geographic information systems can notably enhance intervention accuracy and timeliness for preventing or treating plant diseases.

This section explores the current state of NER applications in plant health domain by examining unique challenges faced, technological approaches

used,and practical implementations of NER systems. Additionally, it looks at potential future advancements and ongoing research aimed at refining these systems to better support complex decision-making needs in modern agriculture.

2.4.1 . Challenges Specific to Plant Health NER

Implementing Named Entity Recognition for plant health introduces a set of unique challenges that underscore the complexities of agricultural language and the specificity of botanical information. Addressing these challenges is crucial for the successful application of NER technologies in enhancing agricultural research and practice.

Linguistic Ambiguity and Terminological Complexity Agricultural texts often contain a significant degree of linguistic ambiguity. Terms like "rust" can refer to a plant disease or simply to oxidized metal, depending on the context. Similarly, "blackberry" might denote a fruit or a mobile device. Such ambiguities necessitate sophisticated NLP systems capable of contextual differentiation. Additionally, the complexity of scientific terminology, including Latin names, like *Erysiphe necator* (a type of fungus) and specialized jargon, like *phytoremediation* (a method that use plants to purify polluted soil), requires an NER system to have an extensive and continually updated vocabulary to recognize and differentiate these terms accurately.

Data Scarcity and Lack of Annotated Datasets The field of plant health does not have as extensive a repository of annotated datasets as more general domains like news or popular domains like finance or healthcare. The scarcity of labeled data in agricultural contexts hampers the training of effective machine learning models. Generating these datasets is often costly and time-consuming, as it requires domain expertise not only in linguistics but also in various agricultural disciplines.

Diverse Data Sources Information pertinent to plant health is disseminated through various channels ranging from academic articles and technical reports to social media posts and direct farmer communications. Each of these sources has its own linguistic style and technical complexity, challenging NER systems to maintain consistency and accuracy across different types of texts.

Fast Evolution of Domain-Specific Knowledge The fields of botany and agriculture are continually evolving, with new pest species being discovered and new diseases emerging. NER systems need to be adaptive

Dataset	Entity types	Mentions number	Documents number	Document types	Availability
Plant-Disease Relations	Plant, Disease	3160	199	Abstracts	doi.org
Plant corpus	Plant	3985	208	Abstracts	gcancer.org
PPR	Plant, Phenotype	16937	600	Abstracts	doi.org
Agricultural corpus	Person, Location, Organization, Chemicals, Crop, Organism, Policy, Climate, Food items, Diseases, Natural Disaster, Events, Nutrients, Count, Distance, Quantity, Money, Temperature, Date	11041	-	Wikipedia articles, websites	-
Plant-Chemical Relationships	Plant, Chemical	742	382	Abstracts	-
Taec	Species, Trait, Phenotype	10815	528	PubMed Articles	doi.org

Table 2.1: **Summary of datasets related to Plant health monitoring and agriculture for NER relevant to our study.** This table includes details on the number of mentions and documents, types of documents, and entities types for each dataset.

and quickly updatable to incorporate new terminologies and entities as the domain knowledge expands.

Moving from the complexities inherent in plant health language to practical approaches for handling these challenges, we transition into discussing the actual data sources and methodologies used in this study for Named Entity Recognition in plant health.

2.4.2 . Data Sources and Annotation for Plant Health NER

In this study, we focus on four entity types: plants, pests, locations, and diseases. These categories are fundamental to the study of plant health as they encompass the key aspects of agricultural ecosystems—flora and the threats they face, both biological (pests and diseases) and environmental (locations). Understanding and extracting information related to these entities are vital for effective monitoring and management of crop health. However, data annotated with these specific entity types is scarce. Researchers often develop custom corpora tailored to their specific needs, as seen in [Liang et al., 2023], [Seideh et al., 2016], [Yan and Li, 2021], and [Liu et al., 2020]; however, these resources are frequently not made publicly available, maintaining the data scarcity issue. This section details all relevant publicly available datasets that have been identified to address this gap. A summary table of these datasets are provided at Table 2.1 and Table 2.2.

Plant corpora

The Plant-Disease Relations Corpus [Kim et al., 2019] is a resource specifically designed to facilitate text-mining of plant and disease relationships documented in Medline abstracts. This corpus categorizes plant-disease interactions into four distinct types: treatments, causes, associations, and negative relations. However, a significant limitation of this corpus for

our study is the ambiguity between human and plant diseases, which can complicate the extraction and analysis of disease-specific data relevant to our focus on plant pathology. Additionally, the broad interpretation of the term "plant" in the corpus could potentially impact the precision of our analyses.

The Plant corpus [Cho et al., 2017] was constructed for NER/NEL for plants from scientific papers abstracts. The only annotated entity type is Plant. All the mentions were mapped to concepts in the NCBI taxonomy database [Federhen, 2012]. Given its specialized focus on scientific names of plant entities, its utility in our research is limited.

The Plant-Phenotype Relationship (PPR) Corpus introduced in [Cho et al., 2022] is a dataset for Relation Extraction between plants and phenotypes in biomedical literature. This corpus is constructed from manually annotated abstracts sourced from PubMed. It contains annotated mentions of plants and phenotypes along with the relationships between them. This corpus is primarily focused on phenotypes and does not really fit plant health domain.

The agricultural corpus introduced in [Malarkodi et al., 2016], derived from Wikipedia articles and authoritative websites focused on European agriculture, includes 19 annotated entity types. These types represent key terms prevalent in the agricultural domain, such as crop names, diseases, and chemicals. The corpus covers a wide range of agricultural sub-domains, from crop cultivation to agribusiness, ensuring comprehensive coverage of the field's terminology and contexts. This makes it a potentially comprehensive resource for agricultural NER. However, it is important to note that this corpus is not available as open source.

The Plant-Chemical Relationships Corpus [Choi et al., 2016] serves as a valuable resource designed to facilitate the exploration of interactions between plants and chemicals through text mining techniques. Constructed from PubMed abstracts, this corpus includes detailed annotations of relationships between plant and chemical entities. This corpus could be used for LM fine-tuning. However, it is focused on chemicals and their relationships with plants and has a limited usage in plant health domain. Furthemore, although the original publication asserts that the corpus is available online, attempts to access it via the provided link have been unsuccessful.

The Triticum aestivum trait Corpus (TaeC) [Nédellec et al., 2024] was developed to improve the understanding of complex phenotype-genotype relationships and enrich the annotation of diverse trait expressions within agricultural texts. It contains annotations of wheat traits, phenotypes, and species, using the Wheat Trait and Phenotype Ontology for annotations. TaeC is aligned with the D2KAB project, aiming to refine text mining methods and ensure the quality of data in plant genomics research. This corpus provides a valuable resource specifically for wheat species research but does not

Dataset	Mentions number	Documents number	Document types	Area	URL
GeoVirus	2167	229	News (epidemiology)	worldwide	GitHub
LGL	4793	588	Local news	mostly U.S.	*
Clust	11564	1082	news	worldwide	*
GeoWebNews	5121	200	news	worldwide	GitHub
WikToR	25000	5000	Wikipedia articles	worldwide	GitHub
GeoCorpora	2966	211	tweets	mostly North America	GitHub

*This corpus can be found online; however, the original publication does not cite a specific source.

Table 2.2: **Summary of datasets for geographic NER relevant to our study.** This table includes details on the number of entities and documents, types of documents, and geographical coverage for each dataset.

encompass other plant species, limiting its applicability to broader studies.

Overall, the presented corpora are relevant to our research but have limitations due to their focus or unavailability.

Geographical corpora

The geospatial corpora are primarily focused on the recognition and linking of geographical entities. Although these collections do not cover all domain-specific entity types, they are particularly relevant to our research due to the diversity of geospatial entities represented across various news sources and social media platforms. This variability is important as standard NER corpora often repeat the same locations entities, limiting their applicability in dynamic, real-world scenarios.

The GeoVirus dataset [Gritta et al., 2018b] is a specialized open-source test dataset created to evaluate geoparsing capabilities within the context of news events related to global disease outbreaks and epidemics. Constructed from WikiNews articles, this dataset focuses on the geotagging and geocoding of location references within news stories about human diseases. The dataset provide a robust resource for developing and testing geoparsing and geocoding systems in the domain of epidemiological news coverage.

The Local-Global Lexicon (LGL) corpus [Lieberman et al., 2010] is a collection of news articles drawn from the NewsStand system. Specifically designed to tackle the challenges of toponym resolution, LGL emphasizes smaller, geographically dispersed newspapers. While this corpus is larger than GeoVirus, it is focused primarily on U.S. areas and cannot capture the geographic diversity and linguistic variety found in global datasets. This limitation may affect its applicability for our study.

Clust [Lieberman and Samet, 2011] corpus was created to complement LGL. Focusing on larger news stories, contrasts with LGL by including toponyms generally corresponding to larger, more populous locations. Together, these corpora facilitate a comprehensive evaluation of toponym recognition across a spectrum of news sizes and sources. However, both Clust

and LGL primarily feature general news content, which is different from the specialized domain of plant health.

GeoWebNews [Gritta et al., 2018a] is a dataset designed to enhance the evaluation and implementation of fine-grained tagging and classification of toponyms, supporting the development of geotagging and geocoding/toponym resolution applications. This corpus stands as a resource for geographic information retrieval and semantic evaluation tasks in real-world applications. However, like Clust, it is less relevant for our specific focus on plant health due to its general content orientation.

The Wikipedia Toponym Retrieval (WikToR) corpus [Gritta et al., 2018c] is constructed using a Python script that links locations from the GeoNames database to Wikipedia pages. This setup provides a dataset particularly useful for resolving highly ambiguous locations, offering a valuable resource for geospatial entity recognition and disambiguation. Nevertheless, since the corpus is automatically generated from Wikipedia, it may not reflect the full diversity of language found in other domains like news articles or social media.

The GeoCorpora [Wallgrün et al., 2018] is a crowd-sourced and expert-guided corpus which was specifically developed to benchmark and improve geoparsing methods through the manual annotation of location entities in microblog content, particularly from Twitter. Unfortunately, the GeoCorpora is primarily concentrated on North American regions and lacks the geographic diversity and linguistic variety that global datasets offer, which imposes a limitation for our research.

Thus, exploring various corpora for Named Entity Recognition tailored to specific domains reveals that these datasets are highly specialized. Each dataset has its own limitations and is designed for distinct applications. They are often constrained by the scope and nature of the data collected. A common challenge across these datasets is the ambiguity in entity definitions. For example, in some corpora, the "Location" entity might include relative terms such as "South of Montpellier", while others might not. Similarly, the representation of plant names varies; some datasets require inclusion of the word "tree" as in "olive tree", whereas others do not specify this.

For our research in plant health monitoring, we have selected the GeoVirus corpus for identifying "Location" entities because it is specifically designed for tracking global disease outbreaks and pandemics, primarily in human health. This focus is well-aligned with our domain requirements. Additionally, for plant-specific data, we are using the EPOP corpus (see Section 3.1.2), which is specifically crafted for Plant Health Monitoring.

2.4.3 . Technological Approaches and Models

The ChouBERT model [Jiang et al., 2022b], a French pre-trained language model, was designed for monitoring plant health through tweets. The authors address the challenges posed by limited labeled data in extracting meaningful information from social media, specifically Twitter, to enhance epidemiological surveillance of plant health. ChouBERT was shown to effectively identify and classify tweets about plant health threats, demonstrating its utility in detecting both known and novel natural hazards. The model utilizes token-level annotations and benefits from its ability to generalize across different natural hazards with relatively small datasets. This research contributes to precision agriculture by integrating advanced NLP tools to monitor crop health and predict potential outbreaks through the analysis of real-time data from social media.

The AGRONER approach introduced in [Veena et al., 2023], designed specifically for the agricultural domain, which addresses the challenge of extracting domain-specific entities such as diseases, pathogens, pesticides, crops, places, and soil types from text. Leveraging an extended BERT model with Latent Dirichlet Allocation for topic modeling, the system enhances entity recognition by creating domain-adapted tokenizers and vector representations. This system uses global vectors constructed from weighted topic distributions to categorize words into relevant agricultural entities. The model demonstrates significant effectiveness in an unsupervised setting on a custom corpus. However, both the model and the training corpus are not available online, limiting reusability, reproducibility and accessibility.

A KIWINER [Zhang et al., 2022] is a lexicon and attention-based Chinese Named Entity Recognition model for the agricultural domain, particularly focusing on kiwifruit-related entities. KIWINER combines a BiLSTM and a CRF, and introduces some original techniques like out of vocabulary words detection based on statistics, leveraging lexicons for character embeddings adjustments through an attention mechanism and another attention mechanism focusing on long dependencies in a text to improve entity recognition.

Similarly, a BERT-BiLSTM-CRF model has been employed to identify citrus pests and diseases, as detailed in [Liu et al., 2023b]. This model is trained on a corpus constructed from Chinese authoritative texts and websites dedicated to citrus health. The corpus is available upon request.

Two similar approaches were applied to extract information from Chinese texts about apple disease. The first, as detailed in [Guo et al., 2022], integrates dictionaries and similar words into a character-based BiLSTM-CRF model, while the second [Zhang et al., 2023b] uses a BERT-CRF model in a classic way.

Another agricultural NER framework, introduced in [Guo et al., 2021], is designed for recognizing specific diseases, pests, and treatments in Chinese texts. This approach leverages a fine-tuned BERT model combined with

adversarial training to enhance the recognition and generalization capabilities in handling domain-specific texts, which often contain rare entities and terms unique to agriculture. The fine-tuned BERT helps generate context-sensitive embeddings, capturing domain-specific nuances, while adversarial training adds robustness against input variations, improving the model's performance on rare mentions.

The RSA-CANER model [Zhao et al., 2022] employs the ALBERT architecture — a lightweight version of BERT that shares parameters across its layers to reduce computational costs and increase training speed. It incorporates a multi-feature fusion approach that integrates character-level, radical, and stroke features of Chinese characters, capturing detailed semantic and morphological information vital for the accurate classification of entities. This method uses BiLSTM networks to process text sequences, capturing both forward and backward context, and integrates a CRF layer for optimal sequence tagging. Additionally, it employs a multi-head attention mechanism to enhance feature representation and focus on relevant parts of the text.

Thus, we can observe that the existing models within the agricultural NER domain are notably specialized, often targeting specific plants or diseases. Many of these models rely on state-of-the-art supervised learning techniques or remain proprietary, not available as open-source tools. This situation underscores the importance of ongoing research in this area. Enhancing accessibility and broadening the scope of these models could greatly advance our capability to manage agricultural health more effectively and address a wider array of challenges in the field.

2.5 . Conclusion

In conclusion, the application of NER in the domain of plant health presents a complex but critically important challenge within agricultural technology and plant science. By efficiently processing and analyzing textual information, NER plays a pivotal role in identifying key entities such as species, symptoms, pathogens, and diseases from diverse textual sources. The linguistic and contextual challenges in agricultural texts, such as linguistic ambiguity, terminological complexity, polysemy, and homonymy, require advanced systems capable of conducting a deep contextual analysis and continuous adaptation to evolving domain-specific knowledge. This section has explored the technological approaches, challenges, and the current state of NER in the plant health domain, providing insights into both existing capabilities and avenues for future research. As the field progresses, further advancements in NER technologies will undoubtedly unlock new potentials for managing plant health more effectively, leveraging the growing availability

of textual data across digital platforms. Building upon the challenges and opportunities outlined in this discussion, the following chapters will introduce a new approach to NER tailored specifically for the plant health domain.

3 - Language Model Domain Adaptation: A KeyWord Masking method

Masked Language Modeling (MLM) is a crucial step in preparing language models for their final application. At its core, the MLM task consists in training a language model to fill in missing tokens in a passage, given the context provided by the remaining tokens (see Section 2.2.1). This process inherently makes the model capture language syntax, semantics and intrinsic relationships between tokens [Clark et al., 2019a, Hewitt and Manning, 2019, Jawahar et al., 2019, Chang and Bergen, 2024] and also prepares the model for more complex NLP tasks, such as NER.

The primary advantage of MLM, in the context of domain adaptation, lies in its ability of better adjusting the model to specific types of texts, without the need for labeled data. While traditional MLM fine-tuning techniques have already improved performance in many subsequent tasks, we explored ways to refine these methods further, aiming to enhance their effectiveness, in particular for NER task.

This chapter introduces a technique designed to adjust Language Models to specific domains without the need for large datasets. Termed KeyWord Masking (KWM), this method refines the traditional approach to fine-tuning for the MLM task. Unlike standard methods that involve random token masking, KWM guides the selection of tokens for masking based on their relevance to the domain, facilitated by domain-specific lexicons. This strategy aims to prepare the model to be used for a variety of NLP tasks, including but not limited to NER. We hypothesize that by focusing on domain-relevant tokens, an LM can develop a more nuanced understanding of the language characteristics specific to the domain and its entities, thereby potentially improving its performance across multiple NLP tasks.

3.1 . Methodology

As previously discussed, recent studies on few-shot NER have predominantly explored two methodologies: transfer learning and meta-learning.

Our interest was drawn to a transfer learning method tailored for domain-specific data discussed by [Gligic et al., 2020]. This method begins by pre-training word embeddings on a large corpus of texts specific to a domain. Subsequently, the embeddings are fine-tuned using a smaller, labeled NER dataset divided into two stages: the first for training the model on word prediction using the CBOW algorithm (see Section 2.2.4), and the second

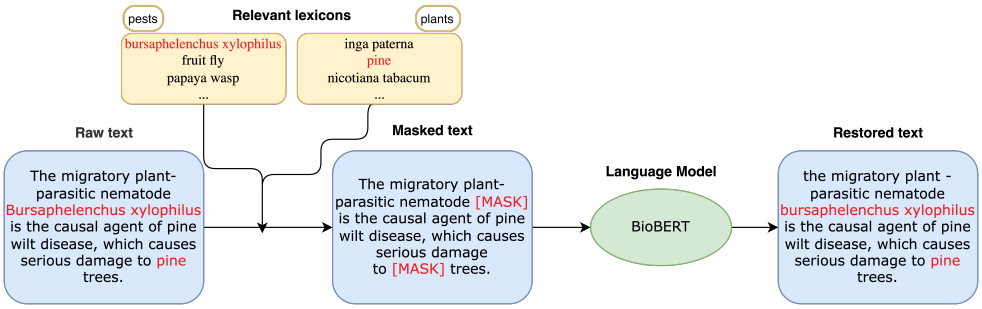

Figure 3.1: **Overview of the KeyWord Masking Strategy.** This schema illustrates the KWM approach to MLM task, where mentions of specific entity types are selectively masked in the text. The Language Model is then fine-tuned to reconstruct these mentions.

specifically for fine-tuning to NER. The distinct advantage of this approach lies in its usage of unannotated texts from the same domain during pre-training, which substantially boosts the NER model's efficacy.

Similar approaches were used by [Pergola et al., 2021] and [Golchin et al., 2023]. In the work [Pergola et al., 2021], the final goal was to improve biomedical question-answering systems. This method enhances language models by selectively masking entities identified by the SciSpacy tool. In contrast, in [Golchin et al., 2023], the masking technique was designed to be task-independent and was tested across several domains. The authors used KeyBERT to extract keywords from datasets and then masked those keywords to fine-tune the LM. It is important to note that although this method was intended to be task independent, its efficacy was only demonstrated in the context of text classification tasks. Even though both of these approaches seem to be efficient, they require a pre-trained NER or keyword extraction system, which could limit their applicability in environments where such resources are unavailable.

Building upon these insights from domain adaptation research, our methodology employs a similar principle of domain relevance but extends it by implementing a KeyWord Masking strategy during the Masked Language Modeling pre-training phase. We assume that this approach prioritizes domain relevance by selectively masking domain-specific keywords, and therefore would enhances the language model's ability to capture nuanced, domain-specific information more effectively. Thus, such targeted adaptation is expected to improve the model's performance on tasks that are sensitive to domain-specific terminology and context.

Unlike previous methods that often depend on pre-existing NER or keyword extraction systems, our KWM strategy operates independently, thus broadening its applicability. We generate and compile lists of domain-relevant terms based on the semantics of entity types that we will further need to

detect. These lists are then used to guide the masking process during MLM domain adaptation. A visual summary of the KWM method is presented in Figure 3.1.

3.1.1 . Masking strategy

As previously mentioned, the Masked Language Modeling task involves restoring masked tokens in a text using the surrounding non-masked tokens. Traditionally, about 15% of tokens are randomly masked, as established in initial research [Devlin et al., 2019]. However, subsequent studies [Wettig et al., 2022, Liao et al., 2022] suggest that masking 40 to 50% of tokens may be more effective. Moreover, tailoring the masking strategy to specific tasks has been shown to yield better results [Lad et al., 2022]. For instance, a strategy presented in [Pergola et al., 2021], as mentioned before, consists in masking tokens identified as entities relevant to the biomedical domain by a NER model, thereby better preparing the language model for biomedical Question-Answering tasks. However, this approach requires a pre-trained NER model. Another interesting approach is [Berend, 2023], where a selective masking technique was employed to concentrate on "important" tokens, while the importance of each token was determined by a task-specific score.

Inspired by these findings, we adopted a similar strategy by masking domain-relevant entities. However, instead of employing a NER system to identify these entities, we compile a list of relevant terms (lexicons) for each entity type (see section 3.1.3) and apply masking using regular expressions (see Figure 3.1). Additionally, recognizing that certain entity types may provide useful contextual clues about others (e.g., predicting a disease from a pest and plant), we ensure that interdependent entities are never masked all together simultaneously. For instance, since certain pests are commonly found on specific plants, masking both could prevent the model to learn these important correlations. Additionally, if the proportion of masked tokens is lower than 15%, we supplement with randomly chosen tokens, taking care not to mask the interdependent entities discussed previously.

3.1.2 . Datasets

To evaluate our method in a robust and generalizable manner, we selected three semantically distinct categories of texts, all in English: Plant Health, Microbiology, and General domain news. The first two categories are highly specialized and distinctly different from each other, though they intersect in the area of microbial diseases. In contrast, the third category includes types of entities frequently encountered in general news articles. We then selected or developed a specific dataset for each category. The evaluation will focus on the impact of KWM on NER, and we will therefore present the corpora and their annotations. This strategy allows us to test our

model under diverse conditions, thereby reducing bias and ensuring fairness in our evaluations.

Plant Health

The entities of interest in the Plant Health domain include *Plant*, *Pest* and *Disease* entities. The primary source of our texts for the dataset that we created and then used is the Plateforme d'Épidémiosurveillance en Santé Végétale (PESV) [PESV, 2023], which provides plant health news summaries, scientific reports, and reports containing updates on plant-related health issues. To enrich this corpus, we supplemented it with a variety of texts, including 578 encyclopedic articles, 23 scientific reports, 102 popular science articles, 61 local news articles, and 48 blog posts. These sources were selected for their rich descriptive content about plant health issues, aligning closely with the operational needs of our real-time NER system. Our selection process involved conducting online searches to identify popular and credible sources, verifying their reliability and ensuring that their licensing terms permitted at least the usage of the material without the possibility of redistribution. These sources include the UK Plant Health Information Portal[1], Phys.org website dedicated to science[2], Encyclopædia Britannica[3] and other. The full list is available at https://github.com/project178/KeyWord-Masking-strategy. Due to licensing restrictions, this corpus is not publicly available. The resulting corpus consists of 1311 texts, all in English, with an average length between 10000 to 20000 characters, thus providing a robust dataset for model fine-tuning in an MLM task.

Subsequently, we selected a subset of this corpus and annotated it for the NER task to obtain preliminary results with "Pest", "Plant" and "Disease" entities. The chosen documents include official reports and news articles that describe pest occurrences on specific plants in certain geographic areas. All the information about the texts can be found at https://entrepot.recherche.data.gouv.fr/file.xhtml?persistentId=doi:10.57745/XTQPNY&version=2.0. To select these texts, we consulted with Plant Health experts from the European and Mediterranean Plant Protection Organization (EPPO). They provided a list of currently monitored pests, which guided our selection of articles that mention all listed pests under various names, to ensure comprehensive coverage.

During the annotation process, we developed specific guidelines, available at https://entrepot.recherche.data.gouv.fr/file.xhtml?persistentId=doi:10.57745/98DYFQ&version=2.0. We used the BRAT format and annotation tool described in the Section 2.3.4. An example of the

[1] https://planthealthportal.defra.gov.uk
[2] https://phys.org/
[3] https://www.britannica.com

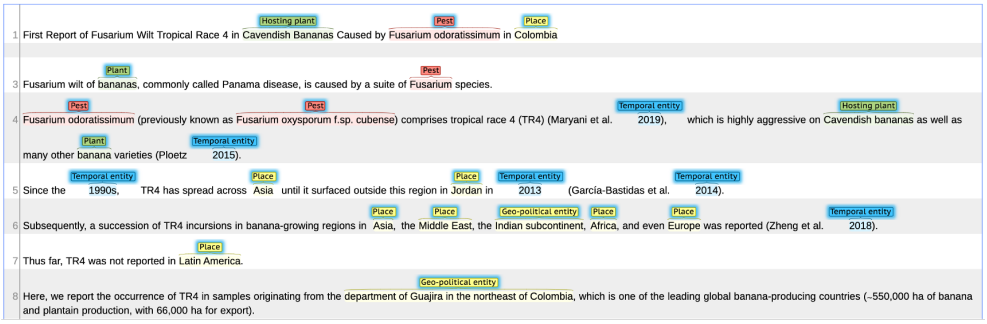

Figure 3.2: **Example of BRAT annotation from our test corpus for named entity recognition in the plant health domain.** Various entity types are annotated, including plants (e.g., "Cavendish Bananas"), pests (e.g., "Fusarium oxysporum"), and locations (e.g., "Asia", "Latin America").

annotation can be seen in Figure 3.2. This corpus is used as a test corpus, and we ensured there were no duplicate documents between our training and testing sets. The annotations of this corpus [Borovikova, 2023] are accessible via the following link[4] however, the text files themselves are not available due to licensing restrictions.

Furthermore, the validation of our approach has been extended to the recently developed Epidemiomonitoring Of Plant (EPOP) corpus, which is set to be released soon. This corpus was specifically designed for Named Entity Recognition, Entity Linking, and Relation Extraction tasks within the plant health domain. It comprises a diverse collection of scientific and media articles, annotated by a team of plant health experts, ensuring its reliability and high quality as a resource for NER applications in plant health. The EPOP corpus contains eight annotated entity types: Pest, Plant, Disease, Vector, Dissemination pathway, Quantity, Date and Location. While the majority of the articles are in English, a portion includes texts in other languages that have been translated using Google Translate, broadening the scope of the dataset.

Microbiology

In the domain of Microbiology, we use the Bacteria Biotope 2019 corpus [Bossy et al., 2019], which includes annotations of *Microorganism*, *Habitat* and *Phenotype* entity types. This corpus comprises abstracts from PubMed about microorganisms and excerpts from scientific articles on beneficial microorganisms in food products. Although the corpus is annotated for several tasks including NER, we use only the raw texts without annotations to fine-tune a language model specific to this domain. For evaluating model

[4] https://doi.org/10.57745/HVPITE

performance, however, we employ both the raw texts and NER annotations from the development set of this corpus. The test set was not used as it is not publicly available.

General-domain news

Unlike microbiology and plant health, the general domain provides a vast variety of existing corpora, allowing us to test the robustness of our approach using different corpora for training and testing. Within this domain, we specifically focus on *Location* entities due to the relative ease of acquiring lexicons. For fine-tuning, we use raw texts from the English CoNLL-2003 corpus [Sang and De Meulder, 2003], which is a subset of the Reuters news stories. To evaluate how well our approach identifies geographical entities, we use the GeoVirus corpus [Gritta et al., 2018b]. This dataset includes 229 news articles, each detailing events related to epidemics and global disease outbreaks.

For a detailed summary of the entity types involved in each test dataset used in our study, refer to Table 3.1. Analysis reveals that the EPOP corpus contains significantly more mentions of pests and plants than diseases, with each plant mentioned approximately five times, pests twelve times, and diseases eight times[1]. In contrast, the Bacteria Biotope corpus exhibits a lower frequency of entity repetition, approximately twice per entity, but habitat mentions are four times more prevalent than phenotype mentions[1]. The Geovirus corpus presents a more challenging dataset, characterized by a high number of unique entities, each appearing approximately three times on average[1]. Consequently, the complexity of identifying and classifying entities may vary, with less various entities potentially having a higher score.

Having constructed a diverse collection of datasets that span multiple domains, we now turn our attention to developing domain-specific lexicons. The construction and implications of these lexicons are discussed in the following section.

3.1.3 . Domain-specific terms lists

In constructing our domain-specific lexicons, we developed separate lists for each type of entity. We chose authoritative sources that are well-regarded in their respective fields. This criterion ensures that our lexicons include all terms necessary for our tasks. Our goal is to use the most representative data sources to improve the accuracy and relevance of our language model across various contexts.

[1]This observation assumes a balanced dataset by entities; however, this balance could be skewed if one or two entities are highly repetitive, overshadowing others that appear infrequently.

Dataset	Documents	Entity type	Number of occurences	Unique Entitites
EPOP	247	Plant	1111	213
		Pest	1202	161
		Diseases	434	50
Bacteria Biotope	134	Microorganism	402	202
		Habitat	610	313
		Phenotype	161	88
Geovirus	229	Location	2167	683

Table 3.1: **Entity Types Across Different Datasets.** This table presents the total and unique counts of various entity types within the EPOP, Bacteria Biotope (development set), and GeoVirus datasets.

Dataset domain	Entity type	Number	Examples
Plant Health	Plant	228984	*baobab, African violet*
	Pest	12107	*oak wilt, pinewood nematod*
	Disease	2844	*HLB, leprosis, apple scab*
Microbiology	Microorganism	6758474	*Psychrobacter immobilis PG1, H. influenzae, Cryobacterium*
	Habitat	4522	*embryonic root part, sesame milk, snow*
	Phenotype	574	*oligotroph, star-shaped, phototactic*
General-domain News	Location	2132976	*Oslo, France, U. S.*

Table 3.2: **Overview of Domain-Specific lexicons for Masked Language Modeling**

The *Plant*, *Pest* and *Disease* entities lexicons are sourced from the EPPO Global database [EPPO, 2023]. It is worth to note that in this database, pests and diseases are combined into a single list named "Pest". To address this, experts from the BEYOND project collaborated with us to clearly differentiate and refine the lists for these two entity types. Managed by EPPO, this database serves as a comprehensive resource containing pest-related data, regularly updated with information produced or collected by EPPO. While it includes many vernacular names, for our purposes, we have retained only those in English.

For the *Microorganism* entity, we used respective subsets of the NCBI taxonomy [Schoch et al., 2020], a structured database that encompasses a wide range of recognized life forms and is constantly updated; however, it includes few vernacular names, which limits its utility. We enriched these subsets by adding automatically generated abbreviations (e.g., *H.equorum* for *Helicobacter equorum*). Lists for *Habitat* and *Phenotype* entities were created by experts.

To mask *Location* entities, we retrieved a list of countries and cities from the GeoNames database [GeoNames, 2023], which catalogs geographic locations worldwide, from major cities to smaller towns. This comprehensive resource was selected for its extensive coverage. Each record in the database contains essential information about the location, including its name in different languages, coordinates, population size, etc.

The gathering of entity-specific terms from these trusted databases form a foundation for identifying domain-relevant entities within the texts, thereby directing the model's focus toward domain-specific vocabulary. This strategic preparation ensures that our language model learns the necessary nuances specific to each specialized domain. An overview of the lexicons is presented in Table 3.2.

3.1.4 . Evaluation method

With the lexicons developed and the fine-tuning phase complete, we now proceed to evaluate the model using two distinct prediction modes. The first involves evaluating the predictions for a randomly selected 15% of the tokens, as is traditionally applied in the literature. The second mode focuses on evaluating the predictions for masked named entities. This involves masking all entities of one type or masking all types simultaneously. To assess the performance, we calculate both accuracy and perplexity, which are standard metrics for the Masked Language Modeling task. Accuracy is calculated as the proportion of correct predictions made by the model out of the total number predictions. Perplexity is calculated by the following formula:

$$Perplexity(M) = \exp(CrossEntropyLoss(M)) =$$

$$\exp(-\sum_{t \in v} L(t|context) * \log_2 P(t|context))$$

where M represents the language model, t denotes a token from the vocabulary v of the model M, $L(t|context)$ indicates the true probability of the token t appearing in the specified context, and $P(t|context)$ is the probability of the prediction of token t by the model M in the given context. Perplexity measures the inverse probability normalized by the number of words in the test set. It effectively quantifies the divergence between the predicted and actual probability distributions of the text. Lower perplexity indicates higher confidence and better model performance, with a score of 1.0 representing perfect prediction. In the study introducing BERT [Devlin et al., 2019], perplexity scores ranged from 3 to 6, which can be considered satisfactory for evaluating this model.

The connection between perplexity and accuracy in evaluating model performance is complex [Gonen et al., 2023]. While accuracy measures the model's precision at predicting each tokens correctly, perplexity evaluates the model's overall confidence across the entire text by analyzing the probability of the generated sequence. As such, these metrics are complementary, offering a detailed and multidimensional evaluation of a model's capabilities and overall effectiveness in the MLM task.

3.2 . Experiments

3.2.1 . Baselines

To assess the effectiveness of our domain-specific fine-tuning approach, we compare it with two baseline models. The first baseline is a model that has not undergone any fine-tuning, operating in its pre-trained state. This serves as a control to demonstrate the model's capabilities without domain-specific adaptations. This comparison helps us determine the extent to which domain-specific fine-tuning provides a measurable improvement over the generic model. The second baseline involves a standard fine-tuning process, where 15% of the tokens are randomly masked. This widely used method aims to refine the model's overall predictive abilities in a specific domain without specifically targeting domain-relevant terms. These comparisons are necessary to determine whether our method improves performance compared to the standard strategy, and if so, to what extent.

3.2.2 . Implementation details

In our experiments, we have fine-tuned BERT [Devlin et al., 2019] and BioBERT [Lee et al., 2020] models. We selected BERT because it is a widely used model accross various domains, and BioBERT, as it is the current State-of-the-art model in the biomedical domain, which includes Microbiology

and Plant Health. Both models were chosen for their ease of use and relatively light computational requirements. While numerous other models are now available and could be tested in future work, the focus of this research is to compare the impact of masking strategies with models that range from general to more specialized domains, rather than surveying all possible models.

For both models training is done with an Adam optimizer and a learning rate $5e-5$. We implemented an early stopping mechanism that stops training if there is no improvement in model performance for five consecutive epochs.

The data and results for this study were processed using Python version 3.8 [van Rossum, 2022]. We primarily used the PyTorch [Imambi et al., 2021] and transformers [Wolf et al., 2020a] libraries for our computational needs. The code is available at `https://github.com/project178/KeyWord-Masking-strategy`.

With the technical setup detailed, we will present the evaluation results of our model in the following section.

3.2.3 . Results

Tables 3.3 and 3.4, along with Figures 3.3, 3.4, 3.5, 3.6, 3.7 and 3.8, show the performance metrics of accuracy and perplexity for the BERT and BioBERT models. These models were tested across various datasets described in section 3.1.2, applying both standard and KeyWord Masking fine-tuning approaches. Details about the type of masking applied during testing are specified in the "Masked tokens" column of each table. The "Dataset" column indicates the datasets used for evaluation, while the "Model" column lists the model used. Results are presented for models without fine-tuning, with standard masking, and with KWM, highlighting the best scores in bold for each dataset and model combination. To confirm our findings were reliable, we averaged the results from ten training iterations and provided the standard deviations.

Our experiments demonstrate significant differences between the two masking methods. Models fine-tuned with KeyWord Masking perform better at identifying domain-specific entities than those fine-tuned with the standard method. In contrast, for random word masking, the standard method outperforms KWM. Additionally, in our tests, BioBERT outperforms BERT on biomedical (Bacteria Biotope) and epidemiological (EPPO - Plant health) texts, while BERT proved more effective with general domain (GeoVirus) texts. These findings indicate that the choice of framework and masking strategy should be carefully considered and integrated into the pipeline depending on the solvable task - Named Entity Recognition in our case.

Another observation concerns the perplexity. Our experiments revealed that the standard fine-tuning method results in lower perplexity on the test

Dataset	Masked tokens	Model	Without fine-tuning	Standard masking	KWM
EPOP	Plant	BERT	0.05	0.12±0.01	0.19±0.03
		BioBERT	0.09	0.15±0.02	**0.21±0.02**
	Pest	BERT	0.03	0.08±0.01	0.18±0.01
		BioBERT	0.03	0.09±0.02	**0.21±0.02**
	Disease	BERT	0.04	0.12±0.02	0.10±0.01
		BioBERT	0.06	**0.16±0.02**	0.15±0.01
	All	BERT	0.03	0.09±0.02	**0.21±0.02**
		BioBERT	0.04	0.09±0.02	0.16±0.03
	Random	BERT	0.39	0.46±0.00	0.3±0.02
		BioBERT	0.37	**0.47±0.00**	0.41±0.02
BB	Microorganisms	BERT	0.03	0.04±0.01	0.07±0.03
		BioBERT	0.02	0.04±0.00	**0.08±0.00**
	Habitats	BERT	0.05	0.03±0.00	0.06±0.02
		BioBERT	0.09	0.09±0.02	**0.11±0.01**
	Phenotypes	BERT	0.07	0.02±0.01	0.06±0.00
		BioBERT	**0.11**	**0.11±0.00**	**0.11±0.00**
	All	BERT	0.02	0.02±0.01	0.06±0.01
		BioBERT	0.03	0.03±0.01	**0.08±0.03**
	Random	BERT	0.37	0.38±0.01	0.12±0.00
		BioBERT	**0.46**	0.43±0.01	0.15±0.00
GeoVirus	Locations	BERT	0.08	0.14±0.03	**0.21±0.1**
	Random	BERT	0.30	**0.41±0.03**	0.08±0.00

Table 3.3: **Accuracy Performance Across Datasets.** This table presents the accuracy metrics for BERT and BioBERT models tested across various datasets for MLM task. It highlights the model performance under no fine-tuning, standard masking, and KeyWord Masking conditions, with the best scores marked in bold for each dataset and model combination. Each row in 'Masked tokens" column indicates which entities were masked during evaluation. Rows labeled "All" indicate that all entity types from the dataset in question were masked, while rows labeled "Random" reflect accuracy when tokens were masked at random, without targeted entity masking. This table provides a comparative view of how each fine-tuning strategy influences model accuracy.

Dataset	Masked tokens	Model	Without fine-tuning	Standard masking	KWM
EPOP	Plant	BERT	953	1.1±0.1	1.3±0.2
		BioBERT	218	**1.1±0.0**	1.1±0.1
	Pest	BERT	1075	1.2±0.1	1.1±0.3
		BioBERT	229	1.2±0.1	**1.1±0.0**
	Disease	BERT	908	1.0±0.1	1.0±0.2
		BioBERT	215	1.0±0.1	**1.0±0.0**
	All	BERT	1212	2.7±0.2	5.1±2.1
		BioBERT	969	1.5±0.2	**1.3±0.3**
	Random	BERT	1720	1.9±0.2	2.7±1.1
		BioBERT	305	**1.8±0.0**	1.8±0.1
BB	Microorganisms	BERT	667661	1.8	14.9
		BioBERT	20709	**1.6**	3.1
	Habitats	BERT	609785	2.0	15.3
		BioBERT	27489	**1.6**	3.0
	Phenotypes	BERT	535986	**1.2**	14.3
		BioBERT	28482	**1.2**	2.0
	All	BERT	26170240	6.2±3.2	6.7±1.4
		BioBERT	313426	6.3±2.1	**5.9±2.3**
	Random	BERT	2605567	1.9±0.5	3.4±0.4
		BioBERT	61243	**1.7±0.9**	3.1±1.9
GeoVirus	Locations	BERT	2432350	**4.8±3.6**	15±2.7
	Random	BERT	2875939	**7.3±5.2**	20±6.1

Table 3.4: **Perplexity Performance Across Datasets.** This table details the perplexity scores of the BERT and BioBERT models on MLM task under different fine-tuning approaches: without fine-tuning, with standard masking approach, and with KeyWord Masking approach. The metrics are presented across various datasets, highlighting the best performances in bold. Each score represents an average of results from ten training iterations, alongside the corresponding standard deviations.

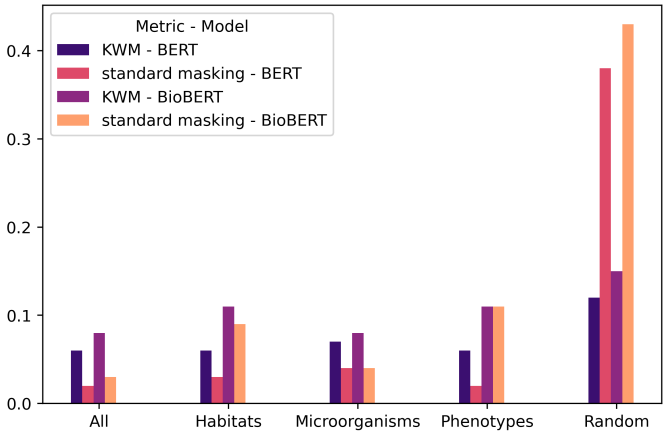

Figure 3.3: **Accuracy comparison of Standard and KWM masking strategies for Bacteria Biotope dataset.**

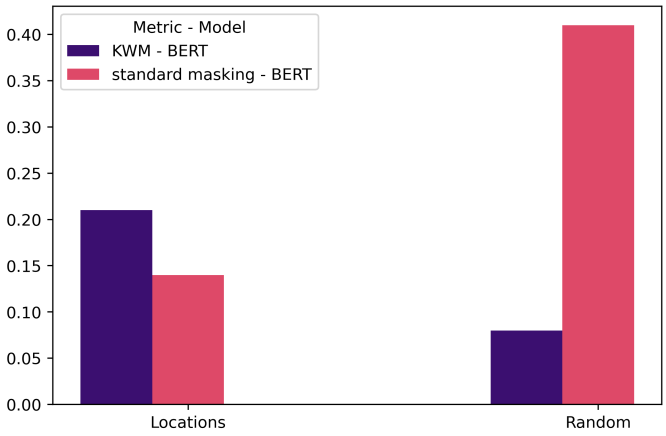

Figure 3.4: **Accuracy comparison of Standard and KWM masking strategies for Geovirus dataset.**

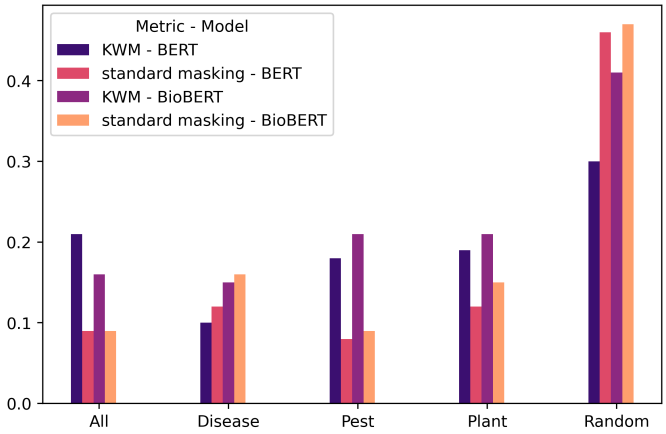

Figure 3.5: **Accuracy comparison of Standard and KWM masking strategies for EPOP dataset.**

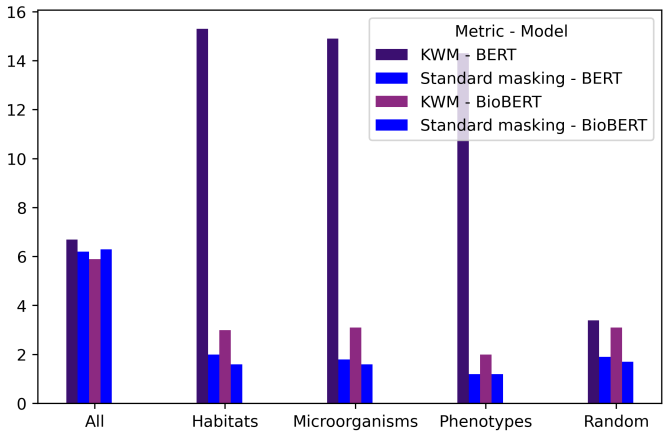

Figure 3.6: **Perplexity comparison of Standard and KWM masking strategies for Bacteria Biotope dataset.**

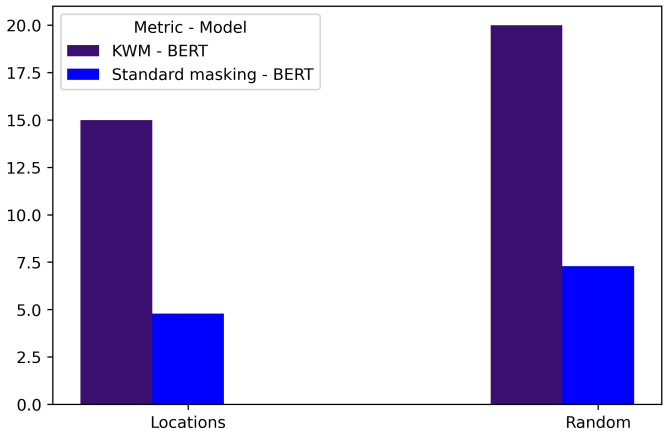

Figure 3.7: **Perplexity comparison of Standard and KWM masking strategies for Geovirus dataset.**

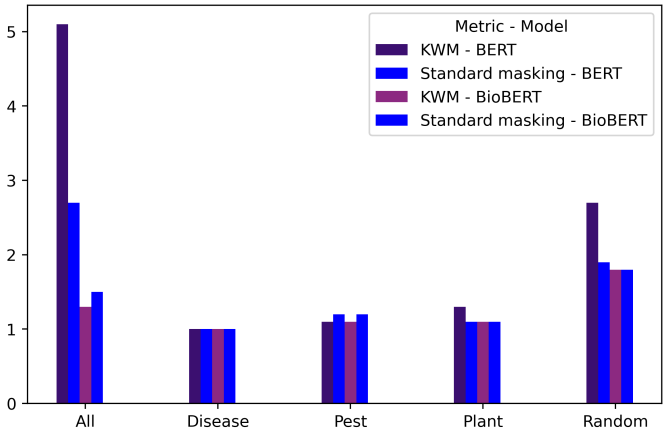

Figure 3.8: **Perplexity comparison of Standard and KWM masking strategies for EPOP dataset.**

set. This suggests that models fine-tuned with the standard approach are more confident in their predictions of masked tokens, highlighting the impact of the choice of the fine-tuning method.

Subsequent tests focus on applying and comparing the pre-trained models to the Named Entity Recognition task. We incorporated a Conditional Random Field layer as a classifier on top of the model, representing a standard architecture for NER (see Section 2.3.5). We used the same models as before, but we eliminated BERTs for EPOP and BB datasets due to lower scores than BioBERT during the MLM evaluation. We used the same datasets previously employed for both fine-tuning and evaluation. Specifically, for the Plant Health domain, where the manually annotated corpus was too small to be used for effective fine-tuning, we used the train and test partitions of EPOP corpus. We computed standard NER metrics such as precision, recall, and F-measure, with the results presented in Table 3.5. The best results are highlighted in bold. As indicated, the models fine-tuned using the KeyWord Masking method generally showed better result. More precisely, the F-score was better for all the entity types, with a significant gap for Plants and Pests entities.

3.3 . Discussion

3.3.1 . General remarks

Overall, our results indicate that the KWM strategy is beneficial for recovering domain-specific terms. The experiments indicate that models fine-tuned using this method grasp the semantics of the masked words' lexical group more effectively, enhancing performance in Named Entity Recognition task involving similar lexical groups. However, this approach appears to diminish the model's general language understanding and offers no benefits when the masked tokens do not belong to a closely related lexical group, as observed in the random token prediction mode.

This makes sense because the standard fine-tuning method involves training the language model on entire text datasets with randomly masked tokens. In contrast, the KeyWord Masking strategy targets training on specific, implicit information that is useful in predicting particular entities. While this technique can enhance the performance of MLMs in tasks like Named Entity Recognition, it may limit the model's ability to capture the general structure and patterns of the texts, potentially reducing its effectiveness in broader processing contexts.

3.3.2 . Entity Type-Specific Observations

In terms of specific entity types, KWM significantly improves accuracy for pests and plants, but shows slightly lower performance for disease entities

Dataset	Masked tokens	Model	Precision	Recall	F
EPOP	Plant	Without fine-tuning	0.55	0.70	0.62
		Standard masking	0.56	**0.78**	0.65
		Keyword masking	**0.59**	0.76	**0.66**
	Pest	Without fine-tuning	**0.68**	0.53	0.59
		Standard masking	0.62	0.49	0.55
		Keyword masking	0.54	**0.73**	**0.62**
	Disease	Without fine-tuning	0.77	0.62	0.69
		Standard masking	0.71	**0.67**	0.69
		Keyword masking	**0.89**	**0.67**	**0.76**
BB	Microorganisms	Without fine-tuning	0.79	0.63	0.70
		Standard masking	**0.80**	0.62	0.69
		Keyword masking	0.77	**0.65**	**0.71**
	Habitats	Without fine-tuning	**0.64**	0.43	0.51
		Standard masking	0.59	0.43	0.50
		Keyword masking	0.60	**0.46**	**0.52**
	Phenotypes	Without fine-tuning	**0.56**	0.37	0.44
		Standard masking	0.55	0.37	0.44
		Keyword masking	0.55	**0.43**	**0.48**
GeoVirus	Locations	Without fine-tuning	0.80	0.67	0.73
		Standard masking	0.81	**0.70**	0.75
		Keyword masking	**0.85**	**0.70**	**0.77**

Table 3.5: **Named Entity Recognition Performance Metrics.** This table compares the precision, recall, and F-measure of BERT models under three different masking approaches: without fine-tuning, standard masking, and KeyWord Masking. Results are shown for different entity types across three datasets: EPOP, Bacteria Biotope (BB), and GeoVirus. The highest scores for each metric across the testing conditions are highlighted in bold, demonstrating the effectiveness of the masking strategies in varying contexts.

compared to the standard masking method. This discrepancy is likely due to the smaller number of disease instances in the corpus, leading to less training data and reduced generalization for these entity representations. Additionally, the inherent complexity and variability in disease terminology might also limit achieving higher performance levels with the KWM approach.

Location entities are also poorly restored using the KWM strategy for the MLM task, yet this approach enhances performance for NER. Moreover, it requires fewer epochs (2 *vs* 6) to achieve these results. The reason for this is that when tasked with restoring location entities, the model attempts to determine where an event occurred, which is not always straightforward (e.g., "... the form that made him a hero in [LOC] (Turin)."). Through this process, the model learns to discern patterns specific to location entities, even though it may incorrectly predict locations (e.g., "London" instead of "Paris"). This enhances the model's ability to recognize the same named entities during NER

tasks.

Additionally, it was observed that habitats and microorganisms achieve better accuracy scores when using the KeyWord Masking strategy for the masked token restoring task, but this does not extend to the NER task comparing to the standard approach. The explanation is straightforward: the Bacteria Biotope dataset contains a substantial number of habitat and microorganism entities, providing ample data for the model to learn and predict these entities effectively. This allows the model to reach a performance threshold dictated by other factors, such as the model's architecture. In contrast, phenotypes are less represented in the dataset. The KWM strategy compensates for this by focusing more on these infrequently occurring entities, helping to narrow the representation gap.

Another notable observation concerns predicting all entity types simultaneously. It is evident that accuracy significantly declines when entities are masked collectively. One might initially think this is due to a quantitative issue, where masking all entities simultaneously results in fewer token hints for the system. However, it's crucial to recognize that, in our dataset, entities are interrelated. For instance, specific plants are frequently affected by the same pests and diseases. Similarly, microorganisms and phenotypes often correlate, though habitats do so a bit less frequently. Therefore, trying to restore all entities at once is extremely difficult, if not impossible.

3.3.3 . Concerns over Overfitting

Interestingly, random tokens in the Bacteria Biotope dataset are better predicted by a model that was not fine-tuned. This could be because the pre-trained model already has a sufficiently general understanding of the language used in the dataset, making additional fine-tuning less effective or even, as we can see, detrimental by causing overfitting to specific features that do not generalize well. This is particularly plausible since BioBERT, which showed the best results in this case, was originally trained on literature from similar biomedical domains.

3.3.4 . Errors Analysis

Regarding errors, a similar pattern emerges across pests, plants, diseases and microorganisms. Some of the incorrect predictions resemble scientific names but lack sense (e.g., *Colursothtumususylophilus* instead of *Bursaphelenchus xylophilus*). Others merge elements of two distinct terms (e.g., *Fusarium fastidiosa* instead of *Fusarium odoratissimum*, influenced by frequent references to *Xylella fastidiosa* in the texts). Additionally, some predictions contain relevant terms of a different type, such as predicting a disease name, *panama disease*, instead of the actual pest causing it, *Fusarium odoratissimum*.

3.4 . Conclusion

This chapter has detailed an innovative approach to enhancing language model performance through domain-specific adaptations, specifically through the use of KeyWord Masking. This technique represents a shift from conventional random token masking to a more focused strategy that enhances the model's ability to handle specific domain-related tasks, particularly Named Entity Recognition. Our source code and resources are available at https://github.com/project178/KeyWord-Masking-strategy. The results presented in this chapter have been published at the 2023 International Conference on Applications of Natural Language to Information Systems [Borovikova et al., 2023].

Our exploration into domain-specific fine-tuning has demonstrated that while KWM offers substantial improvements in domain-relevant contexts, it may somewhat narrow the model's general language understanding capabilities. This approach is better in detailed, contextual understanding of domain-specific terms, boosting NER performance when the entities are closely related to the targeted domain.

However, the strategy also presents limitations, particularly in handling tasks where general language comprehension is crucial. Our findings suggest that while KWM enhances model performance on tasks closely aligned with the training domain, it does not universally enhance the model's overall language processing abilities across unrelated domains. Nevertheless, further empirical testing is suggested to validate these findings comprehensively.

Future work will aim to generalize and potentially automate the strategy for selecting lexicons and other resources when adapting the language models for various domains (see Section 6.2). This effort will involve gathering and refining documents and lexicons, followed by their integration into the system to enhance adaptability and accuracy.

In conclusion, the adoption of KWM should be considered carefully with respect to the specific requirements and constraints of the task at hand. KWM is particularly beneficial for tasks focused on specific entities or semantic groups, such as Named Entity Recognition, Entity Linking, or Information Extraction. However, it may be less suitable for tasks requiring a broader general understanding of language, such as text generation, summarization, or classification. Ultimately, KWM is best suited for use cases where domain-specific, and more precisely entity-specific, accuracy is prioritized over general language understanding. As language models continue to evolve, the integration of such domain-adaptive techniques will be crucial in advancing the frontier of machine learning applications in natural language processing.

The language model fine-tuned for the Plant Health domain, developed

in this chapter, serves as a base for the entire Plant Health NER system. Moving forward, the next steps involve developing a domain-flexible NER system that can seamlessly integrate with this Plant Health-adapted model. This integration allow the NER system to be better positioned to respond to emerging challenges in disease prevention. We will now proceed to the next chapter, which discusses the development of this NER classifier designed to complement the capabilities of this language model.

4 - Zero-Shot Named Entity Recognition

As previously mentioned, Named Entity Recognition (NER) is a component of Information Extraction systems. It involves identifying text segments, known as mentions, that correspond to predefined categories, known as Named Entity types, which include Person, Location, Organization, Facility, Product, etc. Those segments are called mentions. State-of-the-art methods, primarily based on machine learning, are highly effective but depend heavily on manually labeled data for training. To overcome this limitation, various adaptation techniques have been developed. These include Meta Learning, which implies adjusting model parameters based on the task, and Transfer Learning, which focuses on adjusting model parameters based on the domain of data.

This chapter introduces SemNER, a domain adaptation method that requires minimal annotated data. It uses textual descriptions of entity types as input features and leverages their latent representations to compare with each token in the text and predict rather belonging to the corresponding entity type or not. This approach enables the model to dynamically adapt to data from new domains without modifying its core parameters, thus improving its adaptability across various contexts. The methodological contributions of SemNER are the following: it leverages the semantics of entity types, transfers learned knowledge from source to target domains without extensive labeled datasets, and is evaluated in a zero-shot scenario. By employing these strategies, SemNER improves the adaptability of NER models across diverse contexts and bridges the gap between different domains without altering the core parameters of the model.

4.1 . Methodology

4.1.1 . General Pipeline

As previously discussed in section 2.3.5, traditional domain adaptation techniques in Named Entity Recognition generally focus on the differences between domains, as demonstrated in [Jia et al., 2019]. However, the approach described by [Ma et al., 2022] caught our interest due to its use of label semantics for Named Entity types, which enables the model to understand the contextual meanings of entity types more deeply than traditional methods, which might treat them as arbitrary categories. This method combines the embeddings of the input text with those of the entity types labels through a dot product. Then it makes predictions based on the highest values obtained from this calculation. Entity type

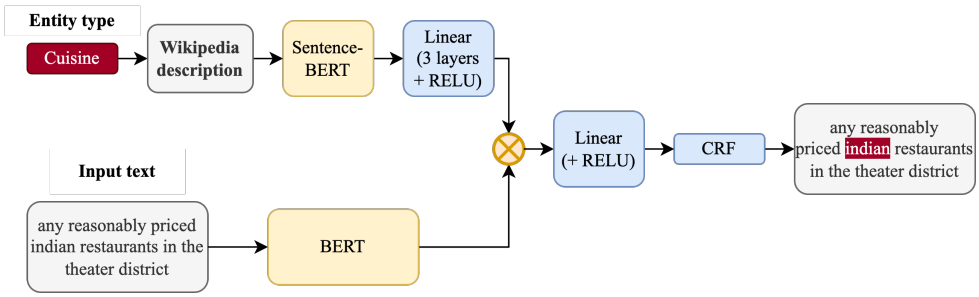

Figure 4.1: **Overview of the Semantic Named Entity Recognition Model.** This diagram illustrates how the model encodes entity types, such as "Cuisine", using Sentence-BERT with descriptions sourced from Wikipedia. Concurrently, input text is encoded using BERT. Both sets of embeddings are processed through linear layers, combined via a dot product, and then passed through a classifier to identify specific entities like "Indian", which is a type of cuisine. The colors in the diagram—yellow for Language Models, gold for the dot product, red for entities, and blue for transformation layers—visually differentiate the stages of data processing.

embeddings are generated by encoding the label names using a separate BERT-based encoder. This process transforms each label, such as "Person" or "Location", into their natural language forms and then produces semantic embeddings for these labels. This strategy differs from conventional methods by leveraging textual representations of entity type labels to directly compare them with the text tokens.

In line with [Ma et al., 2022], we agree that leveraging semantic representations of entity types improves the performance of the NER algorithm while requiring substantially less training data. However, we believe that using only the names of entity types does not fully convey their full semantic meaning. Therefore, building on this concept, we also incorporate label semantics, but we use text passages instead of label names to represent entity types more comprehensively.

Our method starts with the encoding of entity type descriptions and input texts using Sentence-BERT and (Bio)BERT, respectively. These embeddings are then processed through linear layers featuring ReLU activations and combined using a dot product. The subsequent Conditional Random Field layer identifies the matching entities. Figure 4.1 provides a visual overview of our approach.

4.1.2 . Datasets

In our research, we employ datasets from various domains to assess the robustness and broad applicability of our algorithm. We use six datasets in total, some of which have been referenced in the previous chapter: four from the general domain, including the English subcorpus of CoNLL-2003 [Sang and De Meulder, 2003], the GeoVirus dataset [Gritta et al., 2018b], the

Dataset	Entity type	Training data		Testing data		Overlap (raw number)	Overlap (percentage)
		Total	Unique	Total	Unique		
CoNLL-2003	Location	7118	1102	3478	681	379	56
	Organization	6272	2289	2974	1158	535	46
	Person	6560	3485	3423	2226	591	27
	Miscellaneous	3370	735	1593	484	189	39
NCBI diseases	Disease class	769	447	121	84	26	31
	Specific disease	2962	941	555	274	106	39
	Composite mention	115	78	20	19	2	11
	Modifier	1288	271	264	85	51	60
MIT Restaurants	Dish	1475	614	288	168	90	54
	Cuisine	2839	448	532	170	113	66
	Restaurant name	1901	1008	402	291	98	34
	Rating	1070	219	201	73	36	49
	Price	730	180	171	62	33	53
	Hours	990	353	212	122	70	57
	Location	3817	1120	812	337	149	44
	Amenity	2541	1091	533	316	154	49
MIT Movies	Genre	4354	278	1117	132	83	63
	Review	221	99	56	40	16	40
	Rating	2007	48	500	21	17	81
	Ratings average	1869	156	451	83	51	61
	Title	2376	1611	562	448	141	31
	Plot	1927	1174	491	416	191	46
	Director	1719	932	456	315	104	33
	Actor	3219	1285	812	520	377	73
	Character	385	255	90	74	21	28
	Song	245	169	54	46	12	26
	Trailer	113	15	30	6	3	50
	Year	2858	187	720	111	81	73
Bacteria Biotope	Microorganism	739	362	402	202	64	32
	Habitat	1118	579	610	313	36	12
	Phenotype	369	163	161	88	20	23
Geovirus	Location	1733	519	434	186	49	26

Table 4.1: **Datasets Overview.** This table presents counts of total and unique entity types for each dataset, detailed for both training and testing partitions. It also includes counts of overlapping unique entities across these partitions.

MIT restaurant review corpus [Liu et al., 2013b], and the MIT movie review semantic corpus [Liu et al., 2013b]; as well as two from the biological domain, namely the NCBI diseases [Doğan et al., 2014] and the Bacteria Biotope [Bossy et al., 2019] datasets.

Table 4.1 provides a summary of the datasets by entity types, listing the total number of mentions and unique entity designations for each type in the "Total" and "Unique" columns, respectively. It also shows the number of entities appearing in both training and testing datasets ("Overlap raw number" column) and the percentage of these overlapping entities in the test set ("Overlap percentage" column). A higher number of total and particularly unique entities in the training data typically enhances the model's generalization performance and reduces the risk of overfitting. Conversely, a greater number of total and unique entities in the testing data ensures more precise evaluation. Most datasets show a ratio of unique to total entities ranging between 2 and 3 in average. However, some entities are highly repetitive, like *Genre*, *Ratings*, and *Rating average* in the MIT Movies dataset, whereas others like *Title*, *Character*, and *Song* are mostly unique.

The overlap between training and testing datasets also serves as an indicator of the model's generalization capabilities. Specifically, a smaller overlap indicates a more accurate measure of the model's performance on new data. For instance, in the NCBI diseases dataset, the *Composite mention* entity type shows only 2 out of 19 entities common to both the training and testing sets, suggesting minimal overlap. In contrast, the *Cuisine* entities in the MIT Restaurants dataset exhibit about a 66% of test set overlap. These differences are crucial to consider when analyzing the model's performance across various datasets.

Having detailed the diversity and specific characteristics of the datasets employed in our study, it becomes evident why a robust approach to semantic representation is necessary. The variability and uniqueness of entity types across different datasets underscore the need for a method that leverages this diversity for enhanced performance. This leads us into the subsequent section, where we explore how semantic representations of entity types are constructed. By employing an approach that integrates Wikipedia descriptions and advanced semantic processing techniques, we aim to tailor the NER model to effectively handle the complexities introduced by the varied datasets.

4.1.3 . Semantic representation of entity types

This subsection details the methodology for constructing semantic representations of entity types (see Figure 4.1). We begin the process by identifying a suitable textual representation for each entity type. For commonly understood types, such as "Location", we directly use the

corresponding Wikipedia article whose title directly matches the entity type. If a strict match leads to a disambiguation page, we select the relevant article from this page. In cases where the semantic properties of entity type are less straightforward and do not have a correspondent article, such as "date", we select the top result from a Wikipedia search, which in this case might redirect to "calendar date".

To enrich the semantic depth, we also explore articles linked within the primary Wikipedia article. For example, from the article "price", we might extract links to related concepts such as "cost". Additionally, synonyms are incorporated by consulting resources like the Cambridge English Thesaurus [McIntosh, 2023], which enriches the semantic representations of entity types, making them more precise and thereby enhancing the overall quality of the model. This involves matching synonyms with corresponding articles, following the same initial search and selection criteria.

When dealing with ambiguous or unclear entity types, manual selection becomes necessary. For instance, the entity type "genre" in the MIT Movies dataset could ambiguously refer to various contexts; however, we specifically use the "film genre" Wikipedia article to align with the dataset's context. Similarly, for the "org" label in the CoNLL-2003 dataset we select the article for "organization".

Some labels are even more ambiguous. For example, the "Miscellaneous" category in the CoNLL-2003 dataset encompasses a diverse array of names and terms that fall outside persons, locations, or organizations, such as books, movies, conferences, festivals, etc. To address this, we selected an article titled "Named Entity", which provides a several relevant examples.

Another challenging example is the "Modifier" entity type in the NCBI Disease dataset. According to the paper [Doğan et al., 2014], this refers to instances where "the disease mention was not a noun phrase but a modifier". For this, we again used the text from the "Specific Disease" category, concatenating it with text from the "Grammatical modifier" article.

The content from each selected article is then extracted and concatenated into a single document, separated by blank lines to delimit different sources. This document undergoes transformation via Sentence-BERT, producing latent representations of each entity type's textual description.

The latent representations generated are then integrated into the NER pipeline, enhancing the model's capability to recognize and adapt to entity types across various contexts and domains. Figure 4.2 illustrates this integration process and the flow from textual extraction to semantic representation.

With the methodology for constructing semantic representations of entity types, we now proceed to their integration within our Named Entity Recognition model. This integration enhances the model's ability to adapt and

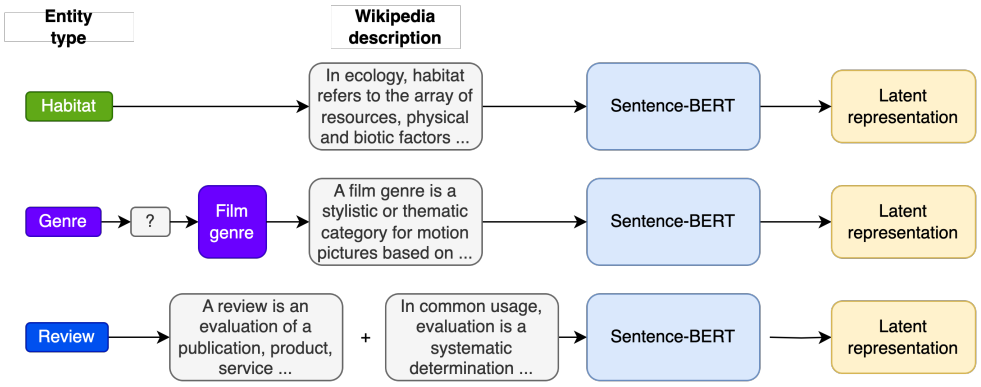

Figure 4.2: **Semantic Representation Process.** This figure illustrates the process of constructing semantic representations for various entity types used in Named Entity Recognition on the example of "Habitat", "Genre", and "Review" entities. The corresponding Wikipedia descriptions are selected for each entity. For ambiguous entity types like "Genre", where the specific domain (e.g., "Film genre") is necessary, a precise article is chosen to match the context of the dataset. Each description is then processed through Sentence-BERT, which transforms the textual content into latent representations.

recognize new or unfamiliar entity types. The next section details how these enriched semantic inputs are incorporated into the model's architecture.

4.1.4 . Model description

In addressing the challenges of Named Entity Recognition, our approach builds upon traditional methods that take tokenized text as input and predict probabilities for each token belonging to predefined entity classes. While these models perform well with familiar data, they struggle to recognize entity classes not encountered during training because they lack the capability to dynamically define new classes. To overcome these limitations of traditional NER models, our method integrates semantic representations of entity types, as detailed in Section 4.1.3. These representations define the entities classes and serve as adaptive cues, enabling the model to handle new or unseen entity types during inference.

We use the BERT [Devlin et al., 2019] and BioBERT [Lee et al., 2020] pretrained models to process input texts. BERT is used for general domain datasets due to its wide applicability across various fields, while BioBERT is selected for biomedical datasets because it is a state-of-the-art model in that area. This choice ensures that the model's language understanding is appropriately aligned with the domain-specific nuances of the input data.

The core peculiarity of our model lies in the integration of embeddings from BERT (or BioBERT) with those from Sentence-BERT [Reimers and Gurevych, 2019]. Initially, a linear transformation layer adjusts the embeddings from both sources to harmonize their dimensions. This

step is necessary as the embeddings from different models reside in distinct feature spaces and cannot be directly combined. After this adjustment, the dot product of embeddings is computed. It measures the similarity between the embeddings, emphasizing features that are common to both, thereby enriching the final feature set used for classification.

These combined embeddings, now enhanced by the dot product, are processed through three subsequent linear layers. These layers progressively refine and reduce the dimensionality of the embeddings, streamlining them for efficient processing and focusing on the most informative features.

The final element in our model's architecture involves a Conditional Random Field (CRF) algorithm (see Section 2.1.1). This layer is particularly valuable as it considers the dependencies between adjacent labels, enhancing the model's precision in identifying the boundaries of named entities.

We maintain the pre-trained weights of both BERT (or BioBERT) and Sentence-BERT fixed, focusing our adaptive training efforts exclusively on the newly added linear layers and the CRF layer. By freezing the base model weights, we ensure the model can be efficiently applied to new datasets without the necessity for extensive retraining, preserving computational resources and enhancing the model's adaptability to evolving data requirements.

An illustrative overview of this model architecture is provided in Figure 4.1.

4.2 . Experiments

4.2.1 . Baselines

To evaluate our approach, we compare it with two baseline models. The first baseline employs cosine similarity - instead of a classifier - between the entity type embedding and each token embedding. We establish a threshold, empirically set at 0.4 after testing various levels during a single epoch of fine-tuning. If the value is below this threshold, the token is classified as belonging to the compared entity type; otherwise, it is classified as not belonging. The second baseline is a BERT model [Devlin et al., 2019] fine-tuned in a classic way. This model does not incorporate the semantics of entity types. Both baseline models, like our own, are trained on the training partitions of all datasets, with one dataset reserved exclusively for testing. These baseline comparisons are essential for understanding the effects of integrating specific semantic information into the model.

4.2.2 . Experimental setup

As detailed in Section 4.1.4, in our study, we employed the BERT [Devlin et al., 2019] and BioBERT [Lee et al., 2020] models for text representation, alongside the Sentence-BERT [Reimers and Gurevych, 2019]

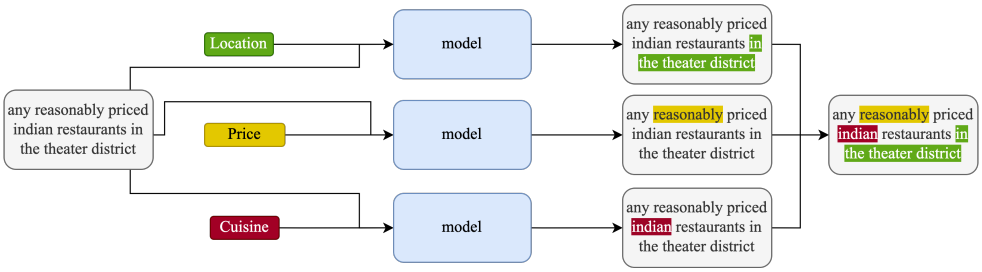

Figure 4.3: **Integration of predictions.** This diagram visually represents the process of consolidating entity predictions for multiple entity types into a unified output. This method enables a holistic evaluation of our model's performance across various entity types.

model for representing entity types. We keep the weights of BERT (or BioBERT) and Sentence-BERT fixed, adjusting only the classifier weights. Given that our model processes one entity type at a time, we generate predictions for each entity within every input text from the test set. We then compile these individual predictions into a unified output that covers all entity types, as shown in Figure 4.3. In cases where a token may fit multiple categories, we assign it to the category with the highest probability. This approach allows for a fair comparison of our model with other existing models.

During the training we used an AdamW [Loshchilov and Hutter,] optimizer and a learning rate $1e-3$. We integrated an early stopping mechanism into our training process, which halts the training if there is no enhancement in model performance observed over five consecutive epochs.

The experiments were conducted using Python 3.10 [van Rossum, 2024], along with the PyTorch [Imambi et al., 2021] and the transformers [Wolf et al., 2020b] libraries.

Having covered the technical setup, we will present the evaluation results of our model in the following section.

4.2.3 . Evaluation method

Having outlined the architecture of our Named Entity Recognition model, we now turn to its evaluation, which was conducted under the zero-shot scenario. Specifically, we trained the model on all datasets except for one, using only the test partition of this excluded dataset for testing purposes.

To assess the model's performance, we calculate precision, recall, and F-measure, which are standard metrics for the NER task.

Precision is calculated as the ratio of tokens correctly identified as entities to the total number of tokens predicted as entities. This metric evaluates the model's ability to label only the relevant tokens as entities.

Recall is defined as the ratio of tokens correctly predicted as entities to

all tokens that are actual entities in the text. This metric assesses the model's ability to identify all relevant instances of entities.

F-measure (also known as F-score or F1-score) represents the harmonic mean of precision and recall. It provides a balanced measure of the model's accuracy by considering both the proportion of correct positive predictions and the relevance of these predictions relative to the actual positives, thus giving an integrated view of performance:

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

While calculating an average of F-measure, there are two approaches: micro-average and macro-average. Micro-average is calculated by summing the individual true positives, false positives, and false negatives of the system for different classes and then using these sums to compute precision, recall, and F1-score. This approach is particularly useful when dealing with imbalanced datasets as it weighs each instance equally, regardless of the class size. Macro-average, on the other hand, involves calculating precision, recall, and the F1-score independently for each class, and then averaging these scores across all classes. This method gives equal weight to each class, regardless of its frequency, which can provide a misleading impression of performance if some classes are much less frequent than others.

In our study, we use micro-average due to the significant class imbalance present in our datasets (see Table 4.1). The micro-average method ensures that our evaluation metrics reflect the performance across all instances in the dataset, preventing larger classes from overshadowing the performance metrics of smaller but equally important classes. By focusing on micro-average, we ensure that our evaluation is not biased by the unequal distribution of classes and facilitate comparison with other studies, as the majority of published research predominantly reports micro-average scores.

The selection of precision, recall, and F-measure as evaluation metrics is particularly appropriate for our study. Precision optimization ensures that the model's performance is not overstated by the number of recognized entities, recall optimization guarantees that the model does not miss relevant entities and F-measure serves as an indicator of overall performance. Together, these metrics provide a thorough assessment of the model's effectiveness across different scenarios.

4.2.4 . Results

This section presents the evaluation of our model. Tables 4.3, 4.4, 4.5, 4.6 and 4.7 present the precision, recall and F-measure scores obtained by our algorithm and two baselines outlined in Section 4.2.1 on MIT Restaurants, MIT Movies, CoNLL-2003 datasets, NCBI diseases and Bacteria Biotope respectively. The "Cosine Similarity" column contains scores from the

Dataset	Standard Cross-Dataset	Cosine Similarity	SemNER	SOTA
CoNLL-2003	0.24	0.12	0.79	**0.93** [Wang et al., 2022]
GeoVirus	**0.79**	0.07	0.54	-
MIT restaurants	0.09	0.12	**0.46**	0.36[Zhou et al., 2023]
MIT movies	0.12	0.02	0.62	**0.63** [Wang et al., 2023b]
NCBI diseases	0.00	0.15	**0.24**	-
BB	0.00	0.17	**0.34**	-

Table 4.2: **Overall comparison of F-score across all the datasets.**

Entity	Measure	Standard Cross-Dataset	Cosine Similarity	SemNER
Dish	Precision	0.00	0.16	**0.38**
	Recall	0.00	0.14	**0.58**
	F-score	0.00	0.15	**0.46**
Cuisine	Precision	0.00	0.14	**0.30**
	Recall	0.00	0.12	**0.86**
	F-score	0.00	0.13	**0.44**
Restaurant name	Precision	0.00	0.08	**0.23**
	Recall	0.00	**0.05**	**0.05**
	F-score	0.00	0.06	**0.08**
Rating	Precision	0.05	0.02	**0.59**
	Recall	0.03	0.12	**0.82**
	F-score	0.03	0.03	**0.69**
Price	Precision	0.00	0.58	**0.82**
	Recall	0.00	0.14	**0.64**
	F-score	0.00	0.23	**0.72**
Hours	Precision	0.00	0.15	**0.44**
	Recall	0.00	0.07	**0.25**
	F-score	0.00	0.10	**0.32**
Location	Precision	0.26	0.16	**0.40**
	Recall	0.32	0.13	**0.84**
	F-score	0.29	0.14	**0.54**
Amenity	Precision	0.00	0.25	**0.85**
	Recall	0.00	0.13	**0.34**
	F-score	0.00	0.17	**0.49**

Table 4.3: **Performance metrics on MIT Restaurants dataset.**

Entity	Measure	Standard Cross-Dataset	Cosine Similarity	SemNER
Genre	Precision	0.00	0.03	**0.98**
	Recall	0.00	0.05	**0.93**
	F-score	0.00	0.03	**0.95**
Review	Precision	0.00	0.00	0.00
	Recall	0.00	0.00	0.00
	F-score	0.00	0.00	0.00
Rating	Precision	0.00	0.00	**0.39**
	Recall	0.00	0.01	**0.15**
	F-score	0.00	0.00	**0.22**
Ratings average	Precision	**0.58**	0.00	0.55
	Recall	**0.82**	0.00	0.80
	F-score	**0.68**	0.00	0.65
Title	Precision	0.00	0.00	**0.15**
	Recall	0.00	0.00	**0.04**
	F-score	0.00	0.00	**0.06**
Plot	Precision	0.00	0.01	**0.21**
	Recall	0.00	0.01	**0.02**
	F-score	0.00	0.01	**0.04**
Director	Precision	0.00	0.03	**0.62**
	Recall	0.00	0.02	**0.89**
	F-score	0.00	0.03	**0.73**
Actor	Precision	0.00	0.05	**0.53**
	Recall	0.00	0.01	**0.72**
	F-score	0.00	0.02	**0.61**
Character	Precision	0.00	0.00	**0.54**
	Recall	0.00	0.00	**0.34**
	F-score	0.00	0.00	**0.42**
Song	Precision	0.00	0.01	0.00
	Recall	0.00	0.00	**0.02**
	F-score	0.00	0.00	0.00
Trailer	Precision	0.00	0.79	**0.98**
	Recall	0.00	**0.91**	0.89
	F-score	0.00	0.85	**0.93**
Year	Precision	0.00	0.02	**0.95**
	Recall	0.00	0.10	**0.25**
	F-score	0.00	0.03	**0.40**

Table 4.4: **Performance metrics on MIT Movies dataset.**

Entity	Measure	Standard Cross-Dataset	Cosine Similarity	SemNER
Location	Precision	**0.93**	0.19	0.91
	Recall	**0.70**	0.15	0.48
	F-score	**0.80**	0.17	0.62
Organization	Precision	0.00	0.22	**0.82**
	Recall	0.00	0.18	**0.90**
	F-score	0.00	0.20	**0.86**
Person	Precision	0.00	0.18	**0.65**
	Recall	0.00	0.04	**0.67**
	F-score	0.00	0.07	**0.65**
Miscellaneous	Precision	0.00	0.00	**0.25**
	Recall	0.00	0.00	**0.11**
	F-score	0.00	0.00	**0.15**

Table 4.5: **Performance metrics on CoNLL-2003 dataset.**

Entity	Measure	Standard Cross-Dataset	Cosine Similarity	SemNER
Disease class	Precision	0.00	0.18	**0.5**
	Recall	0.00	0.23	**0.33**
	F-score	0.00	0.20	**0.40**
Specific disease	Precision	0.00	0.16	**0.55**
	Recall	0.00	0.15	**0.20**
	F-score	0.00	0.15	**0.29**
Composite mention	Precision	0.00	0.01	**0.07**
	Recall	0.00	0.55	**0.65**
	F-score	0.00	0.02	**0.13**
Modifier	Precision	0.00	0.03	**0.10**
	Recall	0.00	0.00	0.00
	F-score	0.00	0.00	0.00

Table 4.6: **Performance metrics on NCBI Diseases dataset.**

Entity	Measure	Standard Cross-Dataset	Cosine Similarity	SemNER
Microorganism	Precision	0.00	0.71	**0.54**
	Recall	0.00	0.14	**0.80**
	F-score	0.00	0.18	**0.75**
Habitat	Precision	0.00	0.12	**0.18**
	Recall	0.00	0.09	**0.13**
	F-score	0.00	0.10	**0.15**
Phenotype	Precision	0.00	0.18	**0.07**
	Recall	0.00	0.15	**0.31**
	F-score	0.00	0.16	**0.12**

Table 4.7: **Performance metrics on Bacteria Biotope dataset.**

cosine baseline model, while the "SemNER" column shows results from our algorithm. The "Standard Cross-Dataset" column reflects scores obtained by a conventional model [Devlin et al., 2019], without incorporating entity type semantics. Notably, a score of zero in this column indicates that the conventional model failed to recognize entities not seen during training, reflecting its limitations in handling unseen data during the inference phase.

Additionally, Table 4.2 presents overall F-scores for all datasets obtained by these models. Scores for SOTA models are provided in the column "SOTA", sourced directly from relevant research papers. We took the best result for each dataset to compare. Specifically, we selected the best published result for each dataset to use as a comparison: kNN-NER [Wang et al., 2022] for the CoNLL-2003 dataset, UniversalNER [Zhou et al., 2023] for the MIT restaurants dataset and InstructUIE [Wang et al., 2023b] for the MIT Movies dataset. It is important to note that no previously published method has been applied to all these datasets simultaneously, making a global comparison impossible. In cases where no score is presented for a dataset, this indicates the absence of zero-shot algorithm evaluations for those datasets up to this point. The scores are intriguing and will be discussed in the Section 4.3.

In evaluating our model across various datasets, we observed a varied performance that reflects both its strengths and areas for improvement. On the MIT Restaurants dataset (see Table 4.3), the model generally performed well across most entities, except for "Restaurant name", which showed weaker results. This may be attributed to the significant semantic variations between the generic concept "Restaurant name" and specific examples such as "Burger King", "Cheesecake Factory", and "Lucky Fortune". These names not only differ widely from the generic concept but also vary considerably among themselves. In the MIT Movies dataset (see Table 4.4), the model scored highly on "Genre" and "Trailer" entities, though it failed to detect

"Review" and "Song" entities almost entirely. The hypotheses attempting to explain these results are discussed in Section 4.3.

On the CoNLL-2003 dataset (see Table 4.5), our model demonstrated strong performance for the "Location" entity, closely matching the precision of the standard cross-dataset model and significantly improving in all scores compared to the cosine similarity model. However, for "Miscellaneous", despite outperforming both baseline models, the scores were still low, highlighting the need for targeted enhancements.

Moving to the NCBI diseases dataset (see Table 4.6), our model delivered acceptable results for "Disease class" and "Specific disease" entities but struggled with "Composite mention" and failed to identify "Modifiers" altogether, indicating areas where the model could be refined. Detailed explanations are provided in the Section 4.3.

Lastly, the performance in the Bacteria Biotope dataset 4.7 was mediocre but still much better than the baselines, indicating a positive direction despite the need for further improvement.

Overall, our experiments demonstrate that our model consistently outperforms baseline models across most entity types, with particularly impressive results in the MIT Restaurants dataset, where it exceeds both the baseline and the current state-of-the-art performance.

However, the model shows weaker results comparing to the standard approach in recognizing the "Location" entity within the Geovirus and CoNLL-2003 datasets and the "Ratings average" entity, indicating areas where further enhancements are needed, which is discussed in the subsequent section.

Having outlined the performance metrics across a range of datasets, we now understand where our model excels and where it requires further development. The following section will delve deeper into the implications of these results, explore the potential reasons behind the disparities in performance, and suggest pathways for future enhancements.

4.3 . Discussion

Our analysis confirms that integrating entity type semantics enables NER model to predict unseen entity types, a capability lacking in standard models, resulting in zero scores for most unseen entities. The only exception is "Ratings average" of the MIT Movies dataset and "Rating" of the MIT Restaurants dataset, but it is worth to note that we allowed a classically fine-tuned model to treat these classes as one class.

A thorough analysis of our prediction results shows clear disparities across different entity types. For instance, entities such as "Location", "Organization", "Person", "Rating", "Price", and "Genre" achieve high scores,

whereas "Miscellaneous", "Modifier", "Restaurant name", "Review", "Title", "Plot", and "Song" score very low. These variations suggest the need for a case-by-case explanation, but broadly speaking, two main factors emerge.

Firstly, the clarity of the concept plays a crucial role. Entities like "Price" and "Location" have straightforward definitions, whereas terms like "Miscellaneous", "Modifier" or "Composite Mention" are inherently vague and confusing even for human annotators. Secondly, the relevance and quality of the source texts or descriptions significantly influence outcomes. For example, while the Wikipedia page for "Song" explains the concept well ("A song is a musical composition performed by the human voice..."), in practice, the system must recognize specific song titles like "Married Life" within context "find me the movie with the song 'Married Life'". This discrepancy indicates that the descriptions often do not align closely enough with the practical identification tasks at hand. It's important to note that these two factors are interconnected: when an entity type is well-defined, it tends to have a shared understanding, which increases the likelihood of having a comprehensive Wikipedia page.

We also can notice that the model is persistently better then the cosine baseline. Therefore, we can conclude that final layers play an important role in retrieving intrinsic features helpful for the classification.

Another observation is that in the case of Cosine Similarity, precision and recall often show similar values. This could suggest several things. It might indicate that the algorithm does not consistently favor one type of error over another—either predicting false positives or false negatives. Alternatively, it could be due to the threshold for classifying an object as part of the positive class being optimized to balance minimizing false positives, which would improve precision, and avoiding missing true positives, which would enhance recall. This optimization can result in comparable values for both metrics. Furthermore, if the model is overly simplistic or lacks the capacity to effectively adapt to the data, such as linear models applied to non-linearly separable data, this too might lead to average yet balanced precision and recall. We believe that the similarity in these metrics likely stems from a combination of the latter two factors: the simplicity of the model and an empirically set threshold that aptly suits the data.

Moreover, it is important to note that for the "Location" entity, a standard model demonstrates lower recall than might be expected given the volume of training data. This discrepancy arises because the definition of "Location" varies significantly across different datasets. For example, in the CoNLL-2003 dataset, "Location" typically refers to simple geographical places like cities and countries. However, in the GeoVirus dataset, it includes more complex geopolitical entities, such as "Fort Hood", or specific sections of facilities, like "St. Nicolas" in "St. Nicolas Hospital". The MIT Restaurants dataset reflects

an even broader approach, incorporating phrases that describe location contextually, such as "around", "in the area", and "near Cheyenne". Despite our model using the semantics of the word "Location", applying a single definition across all these varied contexts leads to the observed lower recall as well. Future work could explore this discrepancy, for example, by training identical models separately on the training partitions of CoNLL-2003 and GeoVirus, then applying them to a mixed set of test data to analyze prediction differences.

Another interesting observation is the confusion between similar classes. For instance, our model frequently predicts the same mention as belonging to both "Disease class" and "Specific disease", or to both "Dish" and "Cuisine" classes simultaneously. We believe this happens because the texts describing these different entities are quite similar. To further investigate this issue, future research could measure the distance between the semantic representations of these similar entity types and compare them with the distances to other, more distinct, entity types.

In addition, it is noteworthy that the "Trailer" entity type is the only one better recognized by the cosine similarity model. The reason is straightforward. According to Table 4.1, the entity "Trailer" appears only 30 times in the test dataset, with 6 unique forms. In fact, over 20 mentions of "Trailer" are represented by the word "trailer" alone. Therefore, adding additional layers for this specific entity type would unnecessarily complicate the model without enhancing performance.

A general trend we have noticed is that categories with fewer representations (refer to the unique and total numbers of entities in Table 4.1) tend to have higher scores. This could suggest that the system is less stable, as its performance drops with a greater variety of mentions. This indicates a potential lack of generalization capability in our model.

Furthermore, it's important to highlight that we compared our zero-shot algorithm with the top-performing zero-shot models for each dataset. Nonetheless, these models did not consistently outperform our approach across different datasets. For example, the study by [Wang et al., 2023b] achieved a score of 0.21 on the MIT Restaurants dataset, which is considerably lower than other reported results. In a similar vein, [Zhou et al., 2023] achieved an F-measure of 0.49 for the MIT Movies dataset, a result that aligns closely with ours. Additionally, all state-of-the-art systems were trained on multiple datasets, often sharing a significant number of common labels, as seen in the CoNLL-2003 dataset. This overlap simplifies the task for these entities, thereby rendering a direct comparison to our method less straightforward and reliable. These observations suggest potential concerns regarding the robustness of existing systems.

4.4 . Conclusion

This chapter has detailed a novel approach on Zero-Shot Named Entity Recognition. By employing entity type semantics integrated by textual descriptions into our model, we have enhanced the ability of NER systems to adapt dynamically to new domains without the need for extensive retraining. Unlike methods that rely on very large models, such as those discussed in [Wang et al., 2023b] and [Zhou et al., 2023], our approach does not require extensive retraining. It also offers ease of transfer to new domains by simply sourcing appropriate textual descriptions, a method that is more straightforward than the model adjustments needed in techniques like those found in [Wang et al., 2022]. The results presented in this chapter have been published at the 27th International Symposium On Methodologies For Intelligent Systems 2024 [Borovikova et al., 2024].

Through testing on diverse datasets, we have demonstrated that our model in some cases surpasses current state-of-the-art approaches, particularly in handling domain-specific entity types. The distinct advantage of using enriched semantic inputs to facilitate domain adaptation has been underscored by our experimental results, which show significant performance improvements, especially in zero-shot learning scenarios where traditional models are less helpful.

However, the variability in performance across different entity types and datasets highlights the ongoing challenges NER systems face, particularly in generalizing across diverse contexts and managing ambiguous entity classifications. The system necessitates meticulous selection of documents that accurately describe the entity types to be identified. The insights gained from this study lay a foundation for future work. Enhancing model sensitivity to context and refining semantic representation sources could all contribute to more robust NER capabilities.

Future research could explore two primary areas:

1. Automating the collection of texts for calculating entity representations (see Section 6.2). One approach to the this problem could involve using dictionaries. For instance, instead of using descriptive texts for the "Location" entity, we could use a list of locations from the GeoNames database, similar to our approach with KeyWord Masking in Chapter 3.

2. Enhancing System Robustness: Developing strategies to make the system both more robust and less sensitive to variations in text is crucial. The specific methods to achieve this remain not clear and require further investigation. However, future research could focus on understanding the factors that contribute to system vulnerability and devising strategies to counteract them. Potential strategies might include experimenting with the Sentence-BERT model to examine

how text variations affect representations. This could involve modifying texts by removing, inserting, or replacing segments and observing the impact on text representations. Additionally, fine-tuning Sentence-BERT for a specific tasks related to entity types, such as predicting entity type based on a text describing it, could provide deeper insights and potential improvements.

Looking ahead, the next chapter will integrate these developments with the other KWM component of the system for domain adaptation in the Plant Health domain. Given the flexibility of this approach, it can be easily applied to specialized domains.

5 - Named Entity Recognition domain adaptation for plant health

As discussed in the previous chapter, adapting a Named Entity Recognition system to a new domain introduces challenges specific to that domain. In the context of plant health, these challenges include managing agricultural and scientific terminology, navigating language ambiguities, and handling the diverse styles of documents, such as academic research papers, technical reports, and social media content. This chapter aims to address these issues by applying two previously presented methods: KeyWord Masking strategy (see Chapter 3) and SemNER (see chapter 4), to effectively adapt NER technology for use in the plant health domain.

5.1 . Methodology

5.1.1 . General pipeline

Building on the insights from previous chapters, this section outlines our methodology for adapting Named Entity Recognition to the Plant Health domain. By integrating two complementary strategies previously discussed, this approach aims to leverage the strengths of each method to enhance the precision and applicability of NER within agricultural texts.

Our method starts by using the KeyWord Masking strategy to fine-tune a domain-specific Language Model. We generate lexicons - lists of pests, plants, diseases, and locations — terms critical to plant health. These lexicons serve as masks during the model's fine-tuning phase. This process helps the model to focus on relevant terms during the Masked Language Modeling task. Afterwards, we use the fine-tuned model to extract latent representations for each token in texts where these entity types are to be detected.

Furthermore, we enhance entity recognition by integrating semantic representations of each entity type. Descriptive articles from Wikipedia about each entity type are processed through the Sentence-BERT model to create rich semantic embeddings. These embeddings are then processed by our SemNER framework through a series of transformations, dot product combination with token representations, obtained by the fine-tuned Language Model, and a Conditional Random Field layer to generate the final predictions.

A visual summary of this integrated approach is provided in Figure 5.1.

5.1.2 . Datasets

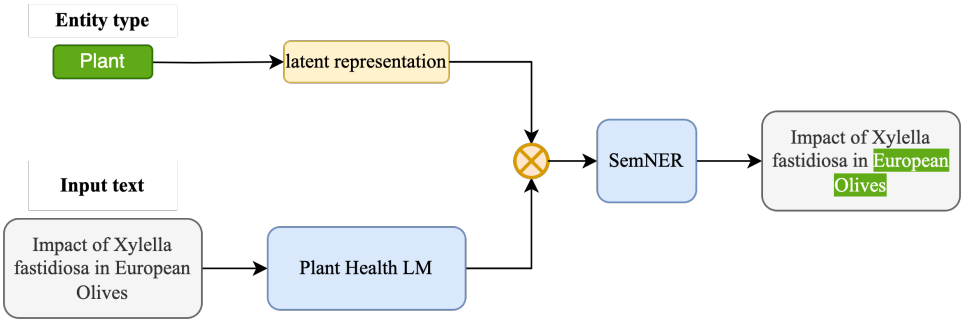

Figure 5.1: **Overview of the Integrated NER Methodology for Plant Health.** This figure illustrates the application of a zero-shot Named Entity Recognition strategy (SemNER), as described in Chapter 4, to the Plant Health domain. It employs a domain-specific Language Model that was fine-tuned using the KeyWord Masking strategy outlined in Chapter 3.

For our experiments we used the two datasets on plant health previously described in Section 3.1.2. More precisely, for the fine-tuning of the Language Model, we employed a collection of documents primarily sourced from the Plateforme d'Épidémiosurveillance en Santé Végétale (PESV), which was further enriched with a variety of related texts, including encyclopedic articles, scientific reports, popular science articles, local news articles, and blog posts, all of which provide rich descriptive content pertinent to plant health.

For the zero-shot NER tests, we leveraged the recently developed Epidemiomonitoring Of Plant (EPOP) corpus, which is specifically designed for several Information Extraction tasks within the plant health domain including Named Entity Recognition. This corpus includes eight entity types (see Section 3.1.2), but our focus was narrowed to the four most critical for our research: *Plant*, *Pest*, *Disease*, and *Location*. This selective focus allows us to concentrate our efforts on the entities most relevant to plant health monitoring. Specifically, understanding *Plant* entities helps in identifying the species at risk or affected by various factors, while *Pest* and *Disease* entities are crucial for diagnosing and managing threats to plant health. Additionally, *Location* information is vital for tracking the spread of pests and diseases and implementing region-specific treatment protocols, making these entities indispensable for effective plant health surveillance and intervention strategies.

In order to describe entity types, we use Wikipedia articles that were specifically selected to capture the essence of each of them, following consultations with Plant Health experts. Specifically, for "Disease" entity type, we use articles on Plant Disease and Plant Pathology. For "Plant", we use Plant and Botany articles, and for "Pest", we use the Pest (organism) article. Additionally, we used the same latent representations for the "Location" entity

as those used in our experiments detailed in Chapter 4.

All the datasets were selected/created to effectively adapt our NER system to the Plant Health domain. However, achieving high performance relies on more than just the dataset. For instance, during the fine-tuning of the Language Model using the KeyWord Masking method, the quality of the lexicons appears to be crucial. We now turn our attention to refining these lexicons used in our experiments, as their precision directly influences the efficiency of the entire system.

5.1.3 . Lexicon filter

While analyzing the results of our KeyWord Masking strategy on entity prediction mode (see Section 3.2.3), we identified that some lexicon entries contained ambiguous short words, leading to false positive predictions. For example, "pit" can refer to a location in Italy as well as a hole in the ground, "rose" can denote both a plant and the past tense of "rise" and "fly" might be identified either as a pest or a verb.

To address this issue, we conducted experiments to determine how many words from our lexicons were simultaneously present in the training dataset and appeared in the testing dataset with a non-relevant meaning. Considering the extensive size of our dictionary, manually analyzing each entry was not feasible. Nonetheless, we had already developed a hypothesis regarding a potential criterion to refine our dictionary. Typically, pests, plants, and diseases are partly denoted by their international scientific names, which are in Latin and tend to be lengthy. Similarly, locations with longer names are less likely to be ambiguous. Based on these observations, we propose that excluding shorter words from our lexicons could effectively reduce these ambiguities.

To test this hypothesis, we projected the lexicon onto our corpus, taking into account the plural forms of the terms. We then assessed the impact of this refinement by calculating precision, recall, and F-measure (refer to Section 2.3.2).

Upon analysis, we determined the optimal length threshold to be different for each category: 11 characters for Locations, 12 for Diseases and Pests, and 4 for Plants. Our primary goal was to prioritize precision over recall, minimising the misidentification of irrelevant tokens. This focus aligns with the KeyWord Masking strategy, which targets specific words rather than random tokens, unlike traditional approach (see Section 3.1.1).

Interestingly, the "Pest" entity exhibited the highest precision at a 30-character threshold, achieving the best precision but with a recall close to zero, indicating a very narrow focus. However, a more balanced approach at a 12-character threshold provided nearly as high precision with a significantly better recall.

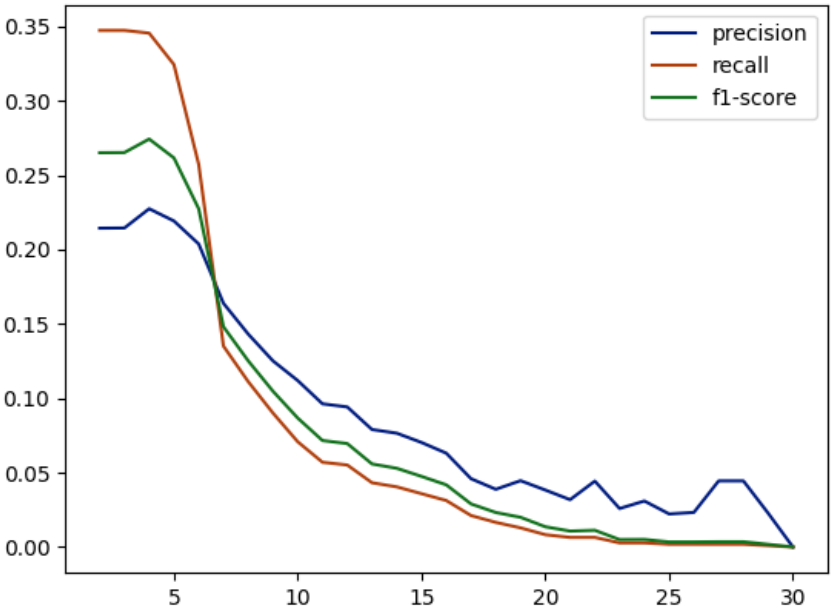

Figure 5.2: **Performance Metrics for "Plant" Entities Using Different Lexicon Length Filters.**

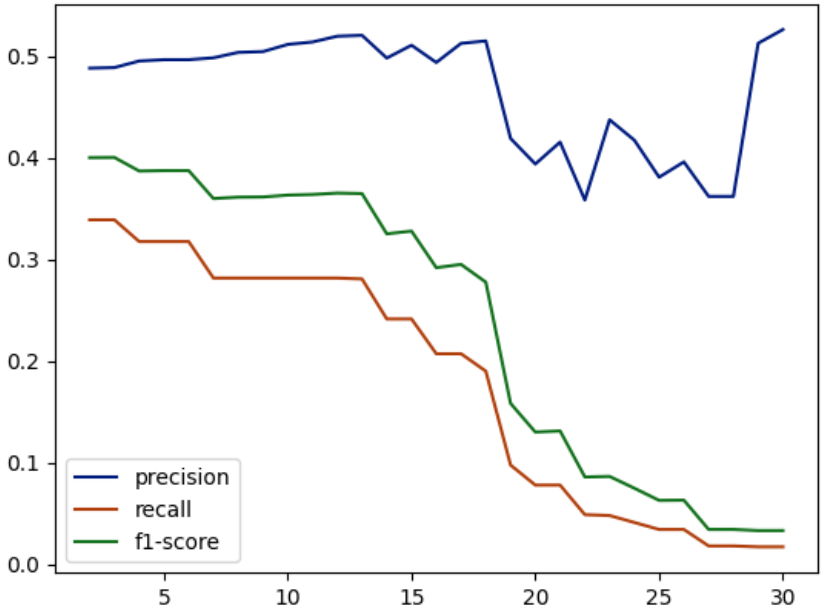

Figure 5.3: **Performance Metrics for "Pest" Entities Using Different Lexicon Length Filters.**

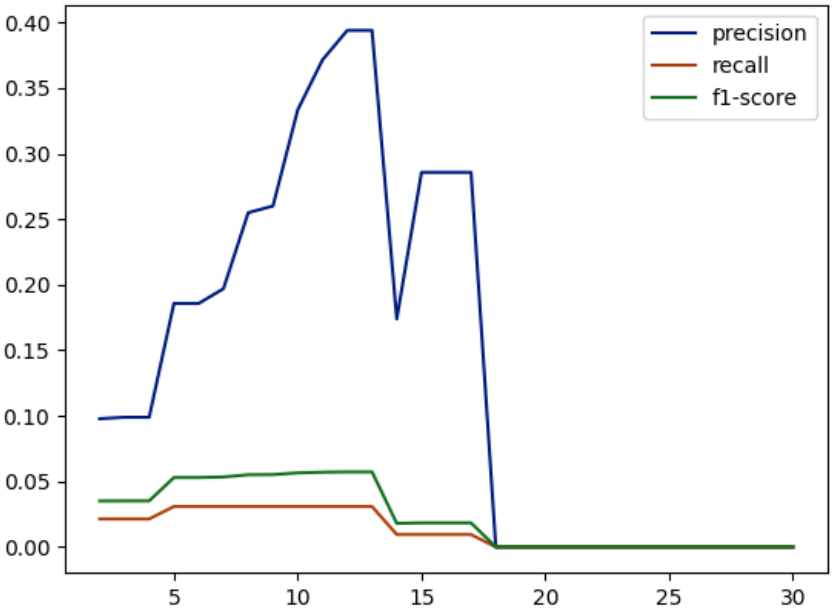

Figure 5.4: **Performance Metrics for "Disease" Entities Using Different Lexicon Length Filters.**

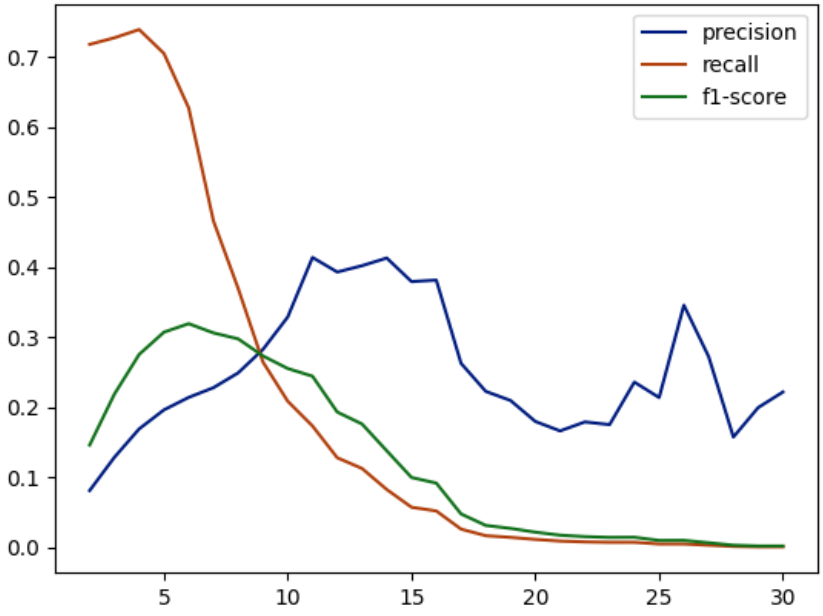

Figure 5.5: **Performance Metrics for "Location" Entities Using Different Lexicon Length Filters.**

The results of these lexicon refinement experiments, demonstrating the impact of different length thresholds on the identification of "Plant", "Pest", "Disease" and "Location" entities, can be viewed in Figures 5.2, 5.3, 5.4, 5.5 respectively.

Having refined our lexicons to exclude shorter, more ambiguous terms, we are now positioned to fine-tune our Language Model more effectively and use these optimized lexicons in a baseline comparison, details of which are described in the following section.

5.2 . Experiments

5.2.1 . Experimental setup

We fine-tuned the BioBERT model [Lee et al., 2020] using the KeyWord Masking strategy following the same protocol as described in Section 3.2.2, and using the filtered lexicons. Subsequently, we apply the SemNER model, developed during the experiments detailed in Section 4.2, to the test dataset. Latent representations of the entities are generated using the Sentence-BERT model [Reimers and Gurevych, 2019]. These representations are derived from texts that were specifically selected to capture the essence of each entity type, following consultations with Plant Health experts.

For the evaluation we use precision, recall and F-measure as described in 4.2.3.

We evaluated our model against three baselines.

1. The first baseline, named "Lexicon projection", uses a regular expression search based on our lexicons, specifically attempting to match each term in both singular and plural forms in the corpus. This baseline provides a clear, low-level solution to compare against more complex models. By establishing how well a simple lexical approach performs, we can evaluate the added value of advanced ML models like BioBERT. In addition, we can evaluate the quality and comprehensiveness of our lexicon.

2. The second baseline employs a standard Transformer-based model, namely BioBERT [Lee et al., 2020] fine-tuned for MLM task with the KWM strategy, enhanced with a Conditional Random Field layer and fine-tuned again for NER task on the train and dev splits of EPOP corpus. This approach combines classic domain-specific supervised task adaptation with the KWM strategy, mirroring the methodology employed in our algorithm. This comparison allows us to assess how our system measures up against traditional supervised domain-adaptation approaches.

Entity	Measure	Standard fine-tuned	Standard cross-datasets	Lexicon projection	SemNER
Plant	Precision	0.56	0.00	0.23	**0.62**
	Recall	**0.78**	0.00	0.35	0.73
	F-measure	0.65	0.00	0.27	**0.67**
Pest	Precision	**0.67**	0.00	0.52	0.18
	Recall	**0.49**	0.00	0.28	0.25
	F-measure	**0.56**	0.00	0.37	0.21
Disease	Precision	**0.73**	0.00	0.39	0.00
	Recall	**0.67**	0.00	0.03	0.00
	F-measure	**0.70**	0.00	0.06	0.00
Location	Precision	0.64	0.65	0.41	**0.75**
	Recall	0.78	0.02	0.17	**0.93**
	F-measure	0.71	0.04	0.24	**0.83**

Table 5.1: **Performance metrics on EPOP dataset**

3. The third baseline is pretrained on the datasets outlined in 4.1.2, but not fine-tuned on the EPOP corpus, serves as a classic supervised system evaluated in a zero-shot setup. This baseline provides a control scenario to evaluate the necessity and effectiveness of domain-specific fine-tuning. If the non-fine-tuned model performs comparably to the fine-tuned models, it might suggest that the pretraining has already captured a sufficient understanding of the relevant concepts, or that the fine-tuning process needs refinement.

All three baselines, along with our model are evaluated on the test split of the EPOP dataset.

5.2.2 . Results

This section presents the results of our experiments. Table 5.1 presents the precision, recall and F-measure scores obtained on the EPOP dataset by our algorithm and by three baselines outlined in Section 5.2.1. The "Lexicon projection" column contains scores from the baseline obtained by matching the lexicons with the corpus (baseline 1), while the "SemNER" column shows results from our algorithm. The "Standard fine-tuned" column reflects scores obtained by a conventional model, fine-tuned on a train split of the dataset (baseline 2). The "Standard cross-datasets" column displays outcomes from a standard model that, like SemNER, is pretrained across various datasets without further fine-tuning on EPOP (baseline 3).

The results indicate a mixed performance across different entity types. Notably, our approach shows promising results in recognizing "Location" entities with significantly higher precision and recall compared to the baselines. However, it struggles with "Disease" entities, where it fails to correctly identify any instances. This disparity will be discussed in the next section. The standard model, when fine-tuned, performs optimally but shows

limited capability in zero-shot mode, recognizing only "Location" entities, which results in zero scores for other entity types. The lexicon projection method, although not achieving top results, still manages to identify 20-40% of entities, except for "Diseases".

The experimental results provide factual insights into the model's performance in generalizing across domains without direct training, highlighting key areas for potential enhancements. As we transition into the discussion section, we will delve deeper into these disparities, examining the underlying reasons for the model's performance variations and exploring strategies for future improvements.

5.3 . Discussion

5.3.1 . Insights from the EPOP dataset

Our results on the EPOP dataset reveal several insights about the performance of our zero-shot NER algorithm, SemNER, within the specific context of Plant Health. Notably, while SemNER demonstrates robust capabilities in recognizing "Location" entities with superior precision and recall, it encounters challenges with "Disease" entities, failing to identify them entirely. This variation in performance across entity types underscores the nuanced challenges inherent in zero-shot learning and highlights the need for targeted improvements.

Firstly, the successful identification of "Location" entities by SemNER suggests that the model effectively leverages semantic features that are generalizable across different contexts. This is likely due to the more defined and consistent representation of location-related terms across various domains and texts. While one might argue that this success is due to prior exposure to "Location" entities during pre-training, it is important to note that two other baselines were also exposed to this entity type. Despite this, SemNER still outperforms the second baseline, which had access to the same datasets. This suggests that SemNER's architecture, which integrates broader contextual understanding, appears particularly suited to such well-defined categories.

Conversely, SemNER's inability to recognize "Disease" entities highlights a critical limitation of the current implementation. This discrepancy indicates a gap in the lexical and semantic coverage of our model, particularly for specialized terms that require a deep domain-specific understanding which our model struggles to capture. Despite various attempts using different textual representations, such as encyclopedic articles, lists of diseases, and related texts, our model consistently failed to detect "Disease" entities. The specialized and varied nomenclatures of diseases within the Plant Health domain pose a particular challenge, as evidenced by the results of the lexicon

projection method in identifying these terms. This result highlights a critical area for improvement in enhancing the model's understanding of complex domain-specific entities.

It's important to note that some diseases are predicted correctly, but their number is so small that it does not affect the overall scores. Namely, 'brown rot' is predicted accurately but only appears twice in the test set. The verification showed that it is mentioned in the Wikipedia article used for entity representation. Surprisingly, 'panama disease', though also mentioned in the same article, was never predicted. When analyzing false predictions, the model often misclassifies pests as both pests and diseases, typically with a higher probability assigned to them being pests. This misclassification could be attributed to the predominance of pest mentions over diseases in the article, which may affect the model's behaviour. However, a similar behaviour is also observed in the conventional model, which sometimes predicts pests instead of diseases, albeit less frequently. This suggests that distinguishing between pests and diseases might generally pose a challenge for NER systems. Indeed, some diseases are named after the pathogens that cause them, such as Fusarium wilt (caused by Fusarium oxysporum) or diseases named directly after viruses like Banana bunchy top virus and Tobacco Mosaic Virus, which refer to both the disease and the causative agent. Further investigation is needed to establish the problem associated with this entity type. Potential experimental approaches to address this challenge are discussed in Section 5.3.3.

Regarding the Standard cross-dataset scores for "Location" entities, we observe a moderately acceptable precision of 0.65, closely aligning with the standard fine-tuning score of 0.64. However, the recall is notably low at 0.02, indicating that while the model does not make a lot of mistakes when it does predict an entity, it fails to detect most entities. A possible explanation for this is that the definition of "Location" varies across datasets, causing inconsistencies in model performance. For instance, the MIT Restaurants dataset primarily contains relative location terms such as "nearby", "around", and "downtown", in contrast to the primarily geographical names found in the CoNLL-2003, Geovirus, and EPOP datasets. Additionally, the CoNLL-2003 dataset sometimes annotates city names within a football context as "Organization" rather than "Location", such as "Paris" and "Saint-Germain" in "Standings in the French first division after Friday's matches (tabulate under played, won, drawn, lost, goals for, against, points): Paris Saint-Germain 21 12 6 3 34 15 42)". When reviewing this model's performance on "Location" entity type across various datasets (see 4.3), it is evident that scores fluctuate depending on the dataset, supporting the hypothesis that the definition of "Location" entities in the EPOP corpus is uniquely distinct, showing minimal overlap with definitions from CoNLL-2003, MIT Restaurants and Geovirus,

which were used for fine-tuning.

Interestingly, the SemNER model outperforms the standard fine-tuning approach in detecting "Location" entities, suggesting several insights into the model's design and application. Notably, the SemNER model's effectiveness suggests that its semantic understanding of "Location" closely mirrors the definitions and expectations set forth in the EPOP dataset. This alignment indicates that the SemNER model has effectively incorporated the specific semantic cues that define "Location" within this dataset, allowing it to recognize and categorize these entities with greater accuracy, while the understanding of "Location" entity of a standard model is dictated by previously seen corpora, which may be different. More precisely, in the EPOP corpus, "Location" typically refers to straightforward geographical places, such as cities and countries (e.g., "Italy"), similar to the usage in the CoNLL-2003 dataset. However, this is in contrast to the Geovirus dataset, where "Location" can include parts of organization titles (e.g., "Bulacan" in "Bulacan Farm"), and the MIT datasets, which describe "Location" in broader terms, such as "nearby".

Additionally, the superior performance of the SemNER model implies that standard fine-tuning methods may face limitations due to insufficient data. Typically, standard approaches rely heavily on the breadth and diversity of training data to learn effective models. However, when the available data does not adequately represent the variety of contexts in which "Location" entities appear, these models struggle to generalize well. This is particularly evident in datasets like EPOP, where unique or less common representations of "Location" may not be sufficiently covered by standard training procedures. For instance, while adjectives are never annotated as "Location" entities in the Geovirus or CoNLL-2003 datasets, in EPOP, adjectives referring to plants, such as "Mediterranean" in "Mediterranean vegetation", are classified as "Location".

Therefore, the success of the SemNER model underscores the importance of tailoring semantic representations to the specific requirements of the dataset and suggests that enhancing training data or employing more sophisticated semantic modeling techniques could significantly improve performance. This analysis sets the stage for a deeper discussion on optimizing entity recognition models to handle varying definitions across datasets effectively.

The results from the lexicon projection method, which achieves moderate success in recognizing a reasonable proportion of entities, further emphasize the potential of incorporating domain-specific lexicons into the learning process. However, the limited success with "Disease" entities also highlights the challenges of the NER task, reinforcing the need for enhanced domain-specific training or the integration of more comprehensive lexicons.

It is worth noting that filtering lexicons had varied impacts depending on the entity type. A common observation is that recall consistently degrades across all types, which is expected as filtering reduces the number of terms available for predictions.

For "Plant" entities, our analysis reveals a notable pattern related to the length of words in our lexicon. As illustrated in Figure 5.2), both precision and recall experience before 4-character filter, indicating the presence of ambiguous words at this length (e.g., "ash", "May", 'tea"). However, a decline in both metrics becomes evident as we increase the lexicon length filter beyond 4 characters. The decrease in recall is expected as fewer words meet the filter criteria, leading to fewer predictions. Interestingly, precision also drops, which can be attributed to the presence of words in the lexicon that appear in the text but are not annotated as "Plant" entities. As the filter becomes more restrictive, excluding more and more relevant words, precision deteriorates. This analysis suggests that words that are 4 characters long strike a balance, being sufficiently relevant and unambiguous for the Plant Health domain.

In contrast, "Pest" entities showed a slight improvement in precision up to a 12-character filter, followed by a a sharp decrease and a subsequent recovery at a 28-character filter (see Figure 5.3). This means that there are some ambiguous terms that have 12-28 characters of length.

For "Disease" entities (see Figure 5.4), precision peaked with a 12-character filter before dropping dramatically to zero at an 18-character filter. Recall slightly improved until until the threshold of 5 characters and then remained stable until a 13-character threshold, beyond which it began to deteriorate, reaching zero at an 18-character threshold. This pattern suggests that there are no long disease mentions in the EPOP corpus that are presented in our lexicon and also that the initial improvement and subsequent stability in recall suggest that shorter terms may not contribute much to false negatives but their elimination eventually limits the model's coverage.

However, the "Location" entities (see Figure 5.5) displayed a notably different trend with sharp fluctuations in precision, peaking and then declining, and recall slightly improving up to a 4-character filter before drastically decreasing. This underscores the complex nature of geographic names and suggests that a more nuanced approach to lexicon filtering may be necessary.

These variations emphasize the need to customize lexicon length filters based on the specific characteristics and requirements of each entity type to optimize the performance of NER systems in different contexts.

In addition, the performance disparities between the fine-tuned standard model and its cross-dataset counterpart highlight the importance of dataset-specific fine-tuning in achieving optimal results. While the standard model fine-tuned on the train split of the EPOP dataset shows respectable

performance, its inability to generalize in a zero-shot scenario without fine-tuning underscores the challenges of applying general NER models to specialized domains without adaptation.

5.3.2 . Potential impact on Plant Health Epidemiology Monitoring

As for the impact on Plant Health Monitoring, considering, this algorithm could be applied in a real world system, as a human assistant, the results are satisfactory or not depending on the particular entity type.

The high precision in identifying "Location" entities ensures that when the model labels a word as a location, it is almost always correct. While high precision minimizes the risk of false positives, which are generally easy for humans to discern, the more critical issue in epidemiological monitoring is the model's low recall. This low recall means that many pertinent locations are not identified, potentially leading to underreported disease spread areas. Such omissions can delay response measures and compromise the overall effectiveness of epidemiological monitoring. Consequently, achieving a recall of 0.93 and a precision of 0.75 is considered a satisfactory outcome, as it balances the need for both accurate and comprehensive location detection.

The failure to detect "Disease" entities significantly impacts epidemiology, as accurate disease identification is crucial for monitoring outbreaks and implementing preventive measures. The low precision and recall in this category mean that the system is currently not reliable for tracking disease occurrences. Even under the best-case scenario, which involved standard training, the highest quality observed was a 70% effectiveness. This level of performance, although substantial, suggests there is considerable room for improvement. High precision is essential to ensure that the diseases identified are correctly classified, avoiding false positives that could lead to unnecessary alarms and misallocation of resources. Similarly, high recall is critical to ensure that all instances of a disease are captured, especially for rare or emerging illnesses that could escalate if not promptly addressed.

Similar to diseases, accurate identification of pests is crucial for effective pest population control and prevention of associated plant health issues. High precision is essential because pests can easily be mistaken for harmless insects or bacteria unrelated to plant diseases. Ensuring that each identified pest is indeed a threat prevents unnecessary interventions and focuses resources on true problems. Similarly, high recall is vital because overlooking a pest can lead to significant delays in taking necessary action, potentially exacerbating the infestation and its detrimental effects on plant health. In the best case scenario we observed, the precision was 0.67 and the recall was 0.49. These figures highlight a substantial gap in the system's effectiveness, underscoring the urgent need for improvements to ensure that pest detection is both accurate and comprehensive.

Additionally, in scenarios where the monitoring focus is on targeted diseases, high precision is better than high recall to ensure that no potential threat is overlooked, which is critical in a surveillance cell monitoring specific diseases. Conversely, less critical diseases, which are more numerous, may be approached more broadly and superficially, prioritizing recall over precision.

Accurate identification of plant species is essential for monitoring disease susceptibility and pest resistance across varied plant types. High precision is crucial, as it ensures that the plants are correctly identified, which supports precise mapping of their susceptibility and resistance. This precision is particularly important given the potential for confusion between plants and closely related entities, such as fruits, herbs, or plant-derived products like medicines. High recall is also vital as it ensures that all relevant plant types are captured in epidemiological assessments, avoiding oversight that could skew data and decision-making.

In our evaluation, the SemNER algorithm achieved the highest precision at 0.62, effectively identifying plants with a lower risk of false positives. On the other hand, the standard fine-tuned model demonstrated superior recall at 0.78, suggesting it is better at capturing a comprehensive range of plant entities but may include more false positives. To leverage the strengths of both approaches, we should explore strategies to integrate the high precision of SemNER with the extensive recall of the standard fine-tuning. Combining these methodologies could potentially yield a more robust system, enhancing both the accuracy and completeness of plant identification in epidemiological studies.

5.3.3 . Future Directions

Moving forward, several strategies could be explored to enhance the performance of zero-shot NER models like SemNER in specialized domains such as Plant Health. One effective strategy involves refining the lexicons; this could be achieved through targeted filtering techniques that eliminate ambiguous or irrelevant terms, or by incorporating synonym expansion to cover a broader range of relevant vocabulary. Such improvements would enable the model to more accurately understand and recognize specialized terms associated with plant diseases.

Furthermore, diversifying the sources and methods for defining entities could offer substantial benefits. Using a broader range of sources enhances models' ability to understand entity types more deeply, potentially increasing accuracy in identifying and categorizing entities, especially in complex or ambiguous contexts. Additionally, diverse sources help mitigate biases inherent in data from a single source. Instead of relying exclusively on encyclopedic articles, which may not capture the nuanced language specific to certain domains, alternative approaches could include:

Expert Input. Engaging domain experts to directly contribute textual definitions or validate existing ones could ensure that the lexicons reflect the precise terminology used in practice.

Usage of Specialized Lexicons. Employing a categorized list of lexicons, where terms are grouped by their relevance or context, might provide clearer guidance for the model on how terms are used within specific domain areas.

Guideline-Based Definitions. Definitions derived from industry or academic guidelines could be incorporated to align the model's understanding with the standardized descriptions recognized in the field.

Particular attention needs to be given to the "Disease" entity type, as the system's behavior here is counterintuitive, necessitating further investigation. As discussed in Section 5.3, we suppose that this may be due to the influence of pest mentions in the articles used to describe the "Disease" entities. This influence could potentially be verified through various text modifications, such as inserting more examples of diseases or removing mentions of pests.

Additionally, challenges in distinguishing between pests and diseases may arise due to similarities in their names.An improvement strategy could involve the integration of a disambiguation module specifically designed to differentiate between closely related terms. This module would not only distinguish between pests and diseases, which are two distinct entity types, but also clarify distinctions between other similar terms, such as plants versus fruits and herbs, or pests versus other insects and microorganisms. Implementing such adjustments would enable the model to refine its recognition strategies dynamically, tailoring its response based on the entity type and the contextual information present.

Another avenue involves incorporating a classic model equipped with an automatic preannotation module. This module would leverage lexicons, particularly focusing on entities that are poorly recognized, such as "Disease" in our study. This preannotation could be used as a silver annotation afterwards. This module would employ lexicons that capture a comprehensive list of disease terms, providing a preliminary layer of annotations. These preliminary annotations, often referred to as 'silver annotations', could serve as an informed guess that the model can refine further.

Moreover, further refinement of the model's architecture to incorporate adaptive thresholds for entity recognition based on the semantic similarity and contextual cues specific to the domain could enhance its adaptability and accuracy.

5.4 . Conclusion

This chapter has described the complex process of adapting Named Entity Recognition for the Plant Health domain through the integration of two sophisticated methodologies — KeyWord Masking and semantic entity representation. Thus, we have crafted an approach to enhance the quality and usability of NER in agricultural contexts.

The methodologies applied here have been rigorously tested using datasets specifically curated for their relevance to plant health. These tests have demonstrated the vital role of accurate, domain-specific lexicons in improving NER performance, highlighted by the significant refinement of lexicons to mitigate the issues posed by ambiguous terms. The experimental results have shown varied success across different entity types, underscoring the nuanced challenges of zero-shot learning and the need for ongoing enhancements to the system's capabilities.

The SemNER model, in particular, excelled in identifying "Location" entities with high precision and recall. Conversely, the struggle with "Disease" entities exposed critical gaps in the model's coverage and its ability to handle the specialized vocabulary intrinsic to plant health. This disparity between entity types has raised discussions on the need for targeted improvements and suggested that future strategies should include the expansion of lexicon sources, the refinement of entity definitions, and the customization of model parameters to better capture the complex dynamics of plant health.

Moving forward, the insights obtained from this study will inform further advancements in NER applications tailored to plant health and related specialized fields. Future efforts will be dedicated to refining the model's capacity to consistently recognize a variety of entity types across different contexts, ensuring high accuracy crucial for practical deployment in plant epidemiology. This will enhance the system's utility in real-world scenarios, enabling more effective monitoring and management of plant health issues.

6 - Conclusion

6.1 . Summary of findings

This thesis has explored the nuances of domain adaptation in Named Entity Recognition focusing on the plant health domain. It introduced novel techniques, namely, a special strategy of Masked Language Modeling with KeyWord Masking and SemNER system which uses semantic representation of entity types as input features. The exploration across the chapters provided deep insights into the complexity of adapting language models to specialized domains, highlighting the effectiveness of targeted strategies in enhancing NER applications.

Firstly, the implementation of the KeyWord Masking strategy, as detailed in Chapter 3, demonstrated how the targeted masking of keywords relevant to specific entities could refine the training process of language models. By focusing on domain-relevant vocabulary, this strategy ensures that the model captures the nuanced context and terminology associated with specific entity types. This entities-oriented approach improved the model's accuracy in recognizing these specific entities such as plants, pests, and diseases within the Plant Health domain, and revealed the importance of domain-specific lexicons in pre-training Language Models for robust NER systems.

This method enhances semantic understanding, enabling the language model to interpret and process text with greater relevance and specificity. As a result, the NER system equipped with this model performs better in real-world applications, especially in identifying entity types that resemble the vocabulary used for masking.

Secondly, the integration of semantic entity representations as described in Chapter 4 offers a method to recognize previously unseen named entities types without requiring an explicit fine-tuning of a model for NER task on the domain-specific dataset. This method leverages textual descriptions of entity types that encapsulate their meaning, thereby enhancing model's adaptability and performance across various domains.

We evaluated this approach on multiple datasets from both general and specialized biological domains. The outcomes highlight our method's broad applicability, as evidenced by non-zero F-scores across all datasets. In addition, our method outperformed several existing zero-shot techniques on certain datasets, achieving an F-score of 0.46 compared to 0.36 [Zhou et al., 2023] and 0.21 [Wang et al., 2023b] on MIT Restaurants, and 0.62 compared to 0.49 [Wang et al., 2023b] on MIT Movies). However, its effectiveness varies and heavily relies on the quality and specificity of the text descriptions used. While this reliance on precise text descriptions could be

seen as a limitation—restricting the ease of application across different entity types — it also presents an opportunity. Since the definition of entity types can vary from one dataset to another and depends on the system's intended purpose, this dependency allows us to fine-tune the algorithm by adjusting the textual definitions of entity types to meet specific needs.

Finally, as outlined in the Chapter 5, the application of a combined strategy in the plant health domain has revealed both the challenges and successes of deploying advanced NER techniques in a specific real-world context, such as plant epidemiology. The high accuracy in identifying location entities contrasted with the struggles in detecting diseases points out critical areas for further research. Identifying "Location" entities within the SemNER model are crucial for tracking the spread of plant diseases and pests. Accurate location data ensures that interventions are correctly targeted, which is vital for efficient resource allocation and response planning.

However, the struggles with identifying "Disease" entities point to significant gaps in the model's ability to handle specialized vocabulary related to plant health. This limitation could influence the detection of disease outbreaks, potentially leading to delayed responses and poor management of plant health crises. Therefore, improving the recognition of disease entities is critical for developing early warning systems and enhancing the predictive capabilities of plant health monitoring systems.

The disparities in performance across different entity types also underscore the need for continuous refinement of the model. For instance, the model's success with "Location" entities suggests that similar strategies could be adapted to improve recognition accuracy for "Disease" and "Pest" entities. Furthermore, the challenges highlighted by the mixed performance across various entity types demonstrate the importance of tailored approaches in NER applications. For effective plant health monitoring, it is essential that the NER system not only recognizes but also accurately categorizes entities such as pests, diseases, and plants in a way that aligns with the specific needs and contexts of the field. This requires an ongoing evaluation and enhancement of the lexicons and training methodologies to ensure they remain relevant and effective.

These insights are crucial for future efforts aimed at refining NER systems not only for plant health but also for other specialized fields that require precise and context-aware text analysis.

In practical terms, integrating these enhanced NER capabilities into real-world plant health monitoring systems could lead to more effective management of plant health issues. While the model fine-tuned in a standard way seems to be more efficient on the same dataset, it can struggle more for new entity types or newly appeared organisms or diseases. That is why, the integration of both, standard and presented system seems the best option

to improve the real-time monitoring systems, enabling proactive measures and timely responses to potential threats. Moreover, the ability to accurately monitor and analyze plant health data on a large scale could transform epidemiological studies, providing deeper insights into disease patterns and pest behaviors, thus contributing to more sustainable agricultural practices and improved food security.

6.2 . Perspectives

Looking ahead, the field of Named Entity Recognition in the context of specialized domains such as plant health offers several avenues for further research and development. The insights gained from this thesis underscore the potential of NER technologies to contribute significantly to domain-specific challenges.

One of potential perspectives is continuous refinement of domain-specific lexicons. Future research should focus on the expansion and precision-enhancement of these lexicons to include more comprehensive and up-to-date terminologies, especially for rapidly evolving fields like plant health. This involves not only enlarging the lexicons but also improving their specificity to reduce the incidence of false positives and negatives.

Exploring further directions in lexicon development, automating the gathering of lexicons through web scraping of specialized websites can efficiently target sites rich in specialized terminology, such as professional associations, industry publications, and academic journals. By automating the extraction of terms and their contexts, systems can rapidly enhance their lexicons with current and relevant vocabulary.

Furthermore, this automation can be configured to regularly update existing lexicons, ensuring that language models remain attuned to the evolving terminology within specialized fields. Such ongoing refinement is crucial for systems operating in dynamic sectors like medical research, legal statutes, technological innovations, or plant health monitoring.

Moreover, employing categorized lexicons, where terms are sorted by relevance or context, can offer clearer insights into the usage of terms specific to certain domains or sub-domains.

Another direction for future research involves refining the methods used to define entities in Named Entity Recognition systems. Relying solely on encyclopedic articles may not capture the nuanced language and specialized terminology of specific domains adequately. To address this limitation, alternative approaches could be pursued to enhance the accuracy and relevance of entity definitions. For example, involving domain experts to provide or verify definitions or documents ensures that the lexicons reflect the specialized terminology used in real-world applications across various

fields accurately. Another strategy could involve using the same lexicons as for KWM directly as text, rather than just definitions, which can further enhance the applicability of the model. Besides, integrating definitions from industry or academic standards can also help align the model's understanding with the commonly accepted terminologies and descriptions used in specialized fields, thereby maintaining consistency and enhancing accuracy in entity recognition across diverse applications.

Another promising avenue for future research is the integration of multimodal data. Studies have shown that combining textual information with visual cues from social media platforms, such as Twitter, can significantly enhance the accuracy of entity recognition [Sun et al., 2021]. This approach uses the relationship between text and images, which could further refine how automated systems gather and interpret complex data across different modalities, thus broadening their application in real-time monitoring and analysis. For example, integrating visual cues from images—such as signs of disease or pest infestation—with geographic data that maps the spread and intensity of these issues, and correlating these with textual analysis from scientific articles, field reports, and social media, could enhance the robustness and accuracy of these systems.

In addition, prompting-based approaches with Large Language Models (LLMs) hold a promise for enhancing NER performance and can be tested. By providing context-specific prompts to LLMs, we can leverage their vast knowledge and language understanding capabilities to refine entity recognition. Recent research, such as [Xie et al., 2024], has demonstrated the effectiveness of this approach by integrating LLMs with modules that automatically generate prompts asking to find NEs in texts. Building on this, we could explore embedding semantic knowledge into LLMs through textual descriptions of named entity types, similar to the descriptions used in our experiments. This approach could potentially further refine NER performance and make it applicable to a specific domain without further adjustment.

Finally, expanding the application of this approach from Plant Health to Human Health and Animal Health could offer several advantages, fostering a more integrated and comprehensive perspective on health and disease management across different domains. This holistic approach aligns with the One Health concept, which emphasizes the interconnectedness of the health of people, animals, and the environment. This could be beneficial in addressing complex global challenges such as climate change, habitat encroachment, and the rising threat of pandemics.

In conclusion, this thesis has contributed to the field of zero-shot Named Entity Recognition within the specialized domain of Plant Health. It introduced two novel approaches: one that fine-tunes a language model more efficiently to a specific domain, and another that applies NER system

to previously unseen entity types. These methodologies were tested on several publicly available datasets from general and biomedical domains, and their integration was applied specifically to Plant Health data. While this combination shows promising results on some entity types, its performance is not consistently stable and varies depending on the representation of entity types. Therefore, while further research is necessary, this work lays a solid foundation for future advancements and practical applications, paving the way for more sophisticated, reliable, and effective NER systems across various specialized domains.

Bibliography

[Achiam et al., 2023] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., et al. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

[Ali et al., 2022] Ali, S., Masood, K., Riaz, A., and Saud, A. (2022). Named entity recognition using deep learning: A review. In *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, pages 1–7. IEEE.

[Amari, 1972] Amari, S.-I. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on computers*, 100(11):1197–1206.

[Amari, 1993] Amari, S.-i. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196.

[Balasuriya et al., 2009] Balasuriya, D., Ringland, N., Nothman, J., Murphy, T., and Curran, J. R. (2009). Named entity recognition in wikipedia. In *Proceedings of the 2009 workshop on the people's web meets NLP: Collaboratively constructed semantic resources (People's Web)*, pages 10–18.

[Baldwin et al., 2015] Baldwin, T., De Marneffe, M.-C., Han, B., Kim, Y.-B., Ritter, A., and Xu, W. (2015). Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the workshop on noisy user-generated text*, pages 126–135.

[Baum et al., 1970] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1):164–171.

[Beltagy et al., 2019] Beltagy, I., Lo, K., and Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620.

[Bengio et al., 2000] Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.

[Berend, 2023] Berend, G. (2023). Masked latent semantic modeling: an efficient pre-training alternative to masked language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13949–13962.

[Bernier-Colborne and Vajjala, 2024] Bernier-Colborne, G. and Vajjala, S. (2024). Annotation Errors and NER: A Study with OntoNotes 5.0. *arXiv preprint arXiv:2406.19172*.

[Bikel et al., 1997] Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a High-Performance Learning Name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201.

[Bischoff et al., 2021] Bischoff, V., Farias, K., Menzen, J. P., and Pessin, G. (2021). Technological support for detection and prediction of plant diseases: A systematic mapping study. *Computers and Electronics in Agriculture*, 181:105922.

[Bojanowski et al., 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

[Borovikova, 2023] Borovikova, M. (2023). Plant Health Risks Identification from textual data. (available online), https://doi.org/10.57745/HVPITE.

[Borovikova et al., 2023] Borovikova, M., Ferré, A., Bossy, R., Roche, M., and Nédellec, C. (2023). Could KeyWord Masking Strategy Improve Language Model? In *International Conference on Applications of Natural Language to Information Systems*, pages 271–284.

[Borovikova et al., 2024] Borovikova, M., Ferré, A., Bossy, R., Roche, M., and Nédellec, C. (2024). Semantically-Informed Domain Adaptation for Named Entity Recognition. In *International Symposium on Methodologies for Intelligent Systems*, pages 55–64. Springer.

[Bossy et al., 2019] Bossy, R., Deléger, L., Chaix, E., Ba, M., and Nédellec, C. (2019). Bacteria biotope at BioNLP open shared tasks 2019. In *Proceedings of the 5th workshop on BioNLP open shared tasks*, pages 121–131.

[Boulaknadel et al., 2014] Boulaknadel, S., Talha, M., and Aboutajdine, D. (2014). Amazighe Named Entity Recognition using a A rule based approach. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 478–484. IEEE.

[Bouri et al., 2023] Bouri, M., Arslan, K. S., and Şahin, F. (2023). Climate-smart pest management in sustainable agriculture: Promises and challenges. *Sustainability*, 15(5):4592.

[Brown et al., 2020a] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020a). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[Brown et al., 2020b] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020b). Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pages 1877–1901.

[Chang and Bergen, 2024] Chang, T. A. and Bergen, B. K. (2024). Language model behavior: A comprehensive survey. *Computational Linguistics*, pages 1–58.

[Chen et al., 2024] Chen, F., Jiang, F., Ma, J., Alghamdi, M. A., Zhu, Y., and Yong, J. W. H. (2024). Intersecting planetary health: Exploring the impacts of environmental stressors on wildlife and human health. *Ecotoxicology and Environmental Safety*, 283:116848.

[Chen and Moschitti, 2018] Chen, L. and Moschitti, A. (2018). Learning to progressively recognize new named entities with sequence to sequence models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2181–2191.

[Chen et al., 2022] Chen, M., Huang, L., Li, M., Zhou, B., Ji, H., and Roth, D. (2022). New Frontiers of Information Extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pages 14–25.

[Cho et al., 2017] Cho, H., Choi, W., and Lee, H. (2017). A method for named entity normalization in biomedical articles: application to diseases and plants. *BMC bioinformatics*, 18:1–12.

[Cho et al., 2022] Cho, H., Kim, B., Choi, W., Lee, D., and Lee, H. (2022). Plant phenotype relationship corpus for biomedical relationships between plants and phenotypes. *Scientific Data*, 9(1):235.

[Cho et al., 2014] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Moschitti, A., Pang, B., and Daelemans, W., editors,

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

[Choi et al., 2016] Choi, W., Kim, B., Cho, H., Lee, D., and Lee, H. (2016). A corpus for plant-chemical relationships in the biomedical domain. *BMC bioinformatics*, 17:1–15.

[Clark et al., 2019a] Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019a). What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

[Clark et al., 2019b] Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2019b). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

[Collobert et al., 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12:2493–2537.

[Cunningham et al., 2008] Cunningham, P., Cord, M., and Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

[Derczynski et al., 2017] Derczynski, L., Nichols, E., Van Erp, M., and Limsopatham, N. (2017). Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Doğan et al., 2014] Doğan, R. I., Leaman, R., and Lu, Z. (2014). Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

[D'Souza, 2024] D'Souza, J. (2024). Agriculture Named Entity Recognition—Towards FAIR, Reusable Scholarly Contributions in Agriculture. *Knowledge*, 4(1):1–26.

[Eftimov et al., 2017] Eftimov, T., Koroušić Seljak, B., and Korošec, P. (2017). A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, 12(6):e0179488.

[Elsayed and Elghazaly, 2015] Elsayed, H. and Elghazaly, T. (2015). A named entities recognition system for modern standard Arabic using rule-based approach. In *2015 First International Conference on Arabic Computational Linguistics (ACLing)*, pages 51–54. IEEE.

[EPPO, 2023] EPPO (2023). EPPO Global Database (available online), https://gd.eppo.int/. Last accessed 06 Feb 2023.

[Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134.

[Federhen, 2012] Federhen, S. (2012). The NCBI taxonomy database. *Nucleic acids research*, 40(D1):D136–D143.

[GeoNames, 2023] GeoNames (2023). GeoNames. https://www.geonames.org/. Last accessed 06 Feb 2023.

[Gligic et al., 2020] Gligic, L., Kormilitzin, A., Goldberg, P., and Nevado-Holgado, A. (2020). Named entity recognition in electronic health records using transfer learning bootstrapped neural networks. *Neural Networks*, 121:132–139.

[Golchin et al., 2023] Golchin, S., Surdeanu, M., Tavabi, N., and Kiapour, A. (2023). Do not Mask Randomly: Effective Domain-adaptive Pre-training by Masking In-domain Keywords. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 13–21.

[Gonen et al., 2023] Gonen, H., Iyer, S., Blevins, T., Smith, N. A., and Zettlemoyer, L. (2023). Demystifying Prompts in Language Models via Perplexity Estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148.

[Grishman and Sundheim, 1996] Grishman, R. and Sundheim, B. M. (1996). Message understanding conference-6: A brief history. In *COLING 1996 volume 1: The 16th international conference on computational linguistics*.

[Gritta et al., 2018a] Gritta, M., Pilehvar, M. T., and Collier, N. (2018a). A pragmatic guide to geoparsing evaluation. *arXiv preprint arXiv:1810.12368*.

[Gritta et al., 2018b] Gritta, M., Pilehvar, M. T., and Collier, N. (2018b). Which Melbourne? augmenting geocoding with maps. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1285–1296.

[Gritta et al., 2018c] Gritta, M., Pilehvar, M. T., Limsopatham, N., and Collier, N. (2018c). What's missing in geographical parsing? *Language Resources and Evaluation*, 52:603–623.

[Guo et al., 2021] Guo, X., Hao, X., Tang, Z., Diao, L., Bai, Z., Lu, S., and Li, L. (2021). ACE-ADP: Adversarial contextual embeddings based named entity recognition for agricultural diseases and pests. *Agriculture*, 11(10):912.

[Guo et al., 2022] Guo, X., Lu, S., Tang, Z., Bai, Z., Diao, L., Zhou, H., and Li, L. (2022). CG-ANER: Enhanced contextual embeddings and glyph features-based agricultural named entity recognition. *Computers and Electronics in Agriculture*, 194:106776.

[Hamosh et al., 2000] Hamosh, A., Scott, A. F., Amberger, J., Valle, D., and McKusick, V. A. (2000). Online Mendelian Inheritance in Man (OMIM). *Human mutation*, 15(1):57–61.

[Hanisch et al., 2005] Hanisch, D., Fundel, K., Mevissen, H.-T., Zimmer, R., and Fluck, J. (2005). ProMiner: rule-based protein and gene entity recognition. *BMC bioinformatics*, 6:1–9.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[Hewitt and Manning, 2019] Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

[Hu et al., 2024] Hu, Z., Hou, W., and Liu, X. (2024). Deep learning for named entity recognition: a survey. *Neural Computing and Applications*, 36(16):8995–9022.

[Huang et al., 2022] Huang, Y., He, K., Wang, Y., Zhang, X., Gong, T., Mao, R., and Li, C. (2022). COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition. In *Proceedings of the 29th International conference on computational linguistics*, pages 2515–2527.

[Imambi et al., 2021] Imambi, S., Prakash, K. B., and Kanagachidambaresan, G. (2021). PyTorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104.

[Iovine et al., 2022] Iovine, A., Fang, A., Fetahu, B., Rokhlenko, O., and Malmasi, S. (2022). CycleNER: an unsupervised training approach for named entity recognition. In *Proceedings of the ACM Web Conference 2022*, pages 2916–2924.

[Jahangir et al., 2012] Jahangir, F., Anwar, W., Bajwa, U. I., and Wang, X. (2012). N-gram and gazetteer list based named entity recognition for urdu: A scarce resourced language. In *Proceedings of the 10th Workshop on Asian Language Resources*, pages 95–104.

[Jawahar et al., 2019] Jawahar, G., Sagot, B., and Seddah, D. (2019). What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

[Jehangir et al., 2023] Jehangir, B., Radhakrishnan, S., and Agarwal, R. (2023). A survey on Named Entity Recognition—datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017.

[Jia et al., 2019] Jia, C., Liang, X., and Zhang, Y. (2019). Cross-domain NER using cross-domain language modeling. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2464–2474.

[Jiang et al., 2022a] Jiang, S., Angarita, R., Cormier, S., Orensanz, J., and Rousseaux, F. (2022a). Choubert: Pre-training French language model for crowdsensing with tweets in phytosanitary context. In *International Conference on Research Challenges in Information Science*, pages 653–661, Cham. Springer International Publishing.

[Jiang et al., 2022b] Jiang, S., Angarita, R., Cormier, S., and Rousseaux, F. (2022b). Named Entity Recognition for Monitoring Plant Health Threats in Tweets: a ChouBERT Approach. In *2022 6th International Conference on Universal Village (UV)*, pages 1–5. IEEE.

[Ju et al., 2011] Ju, Z., Wang, J., and Zhu, F. (2011). Named entity recognition from biomedical text using SVM. In *2011 5th international conference on bioinformatics and biomedical engineering*, pages 1–4. IEEE.

[Jurafsky and Martin, 2018] Jurafsky, D. and Martin, J. H. (2018). N-gram language models. *Speech and language processing*, 23.

[Kim et al., 2019] Kim, B., Choi, W., and Lee, H. (2019). A corpus of plant–disease relations in the biomedical domain. *Plos one*, 14(8):e0221582.

[Kim and Woodland, 2000] Kim, J.-H. and Woodland, P. C. (2000). A rule-based named entity recognition system for speech input. In *Sixth International Conference on Spoken Language Processing*.

[Klampanos, 2009] Klampanos, I. A. (2009). Manning Christopher, Prabhakar Raghavan, Hinrich Schütze: Introduction to information retrieval: Cambridge University Press, Cambridge, 2008, 478 pp, Price 60, ISBN 97805218657515.

[Korkontzelos et al., 2015] Korkontzelos, I., Piliouras, D., Dowsey, A. W., and Ananiadou, S. (2015). Boosting drug named entity recognition using an aggregate classifier. *Artificial intelligence in medicine*, 65(2):145–153.

[Krishnan and Manning, 2006] Krishnan, V. and Manning, C. D. (2006). An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics*, pages 1121–1128.

[Lad et al., 2022] Lad, T., Maheshwari, H., Kottukkal, S., and Mamidi, R. (2022). Using Selective Masking as a Bridge between Pre-training and Fine-tuning. *arXiv preprint arXiv:2211.13815*.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., Pereira, F., et al. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA.

[Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

[Lample and Conneau, 2019] Lample, G. and Conneau, A. (2019). Cross-lingual Language Model Pretraining. *arXiv e-prints*, pages arXiv–1901.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[Lee et al., 2020] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

[Lewis et al., 2020] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[Leydesdorff and Vaughan, 2006] Leydesdorff, L. and Vaughan, L. (2006). Co-occurrence matrices and their applications in information science: Extending ACA to the Web environment. *Journal of the American Society for Information Science and technology*, 57(12):1616–1628.

[Li et al., 2022a] Li, D., Hu, B., and Chen, Q. (2022a). Prompt-based Text Entailment for Low-Resource Named Entity Recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1896–1903.

[Li et al., 2022b] Li, J., Fei, H., Liu, J., Wu, S., Zhang, M., Teng, C., Ji, D., and Li, F. (2022b). Unified named entity recognition as word-word relation classification. In *proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 10965–10973.

[Li et al., 2020] Li, J., Sun, A., Han, J., and Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

[Li et al., 2015a] Li, L., Jin, L., and Huang, D. (2015a). Exploring recurrent neural networks to detect named entities from biomedical text. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 14th China National Conference, CCL 2015 and Third International Symposium, NLP-NABD 2015, Guangzhou, China, November 13-14, 2015, Proceedings 14*, pages 279–290. Springer.

[Li et al., 2015b] Li, L., Jin, L., Jiang, Z., Song, D., and Huang, D. (2015b). Biomedical named entity recognition based on extended recurrent neural networks. In *2015 IEEE International Conference on bioinformatics and biomedicine (BIBM)*, pages 649–652. IEEE.

[Li et al., 2009] Li, L., Zhou, R., and Huang, D. (2009). Two-phase biomedical named entity recognition using CRFs. *Computational biology and chemistry*, 33(4):334–338.

[Li et al., 2021] Li, Y., Shetty, P., Liu, L., Zhang, C., and Song, L. (2021). BERTifying the Hidden Markov Model for Multi-Source Weakly Supervised Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 6178–6190.

[Liang et al., 2020] Liang, C., Yu, Y., Jiang, H., Er, S., Wang, R., Zhao, T., and Zhang, C. (2020). Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1054–1064.

[Liang et al., 2023] Liang, J., Li, D., Lin, Y., Wu, S., and Huang, Z. (2023). Named entity recognition of Chinese crop diseases and pests based on RoBERTa-wwm with adversarial training. *Agronomy*, 13(3):941.

[Liao et al., 2022] Liao, B., Thulke, D., Hewavitharana, S., Ney, H., and Monz, C. (2022). Mask More and Mask Later: Efficient Pre-training of Masked Language Models by Disentangling the [MASK] Token. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1478–1492.

[Lieberman and Samet, 2011] Lieberman, M. D. and Samet, H. (2011). Multifaceted toponym recognition for streaming news. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 843–852.

[Lieberman et al., 2010] Lieberman, M. D., Samet, H., and Sankaranarayanan, J. (2010). Geotagging with local lexicons to build indexes for textually-specified spatial data. In *2010 IEEE 26th international conference on data engineering (ICDE 2010)*, pages 201–212. IEEE.

[Limsopatham and Collier, 2016] Limsopatham, N. and Collier, N. (2016). Bidirectional lstm for named entity recognition in twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145–152.

[Lipscomb, 2000] Lipscomb, C. E. (2000). Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265.

[Liu et al., 2019a] Liu, A., Du, J., and Stoyanov, V. (2019a). Knowledge-augmented language model and its application to unsupervised named-entity recognition. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1142–1150, Minneapolis, Minnesota. Association for Computational Linguistics.

[Liu et al., 2022] Liu, A. T., Xiao, W., Zhu, H., Zhang, D., Li, S.-W., and Arnold, A. (2022). Qaner: Prompting question answering models for few-shot named entity recognition. *arXiv preprint arXiv:2203.01543*.

[Liu et al., 2013a] Liu, J., Pasupat, P., Cyphers, S., and Glass, J. (2013a). Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

[Liu et al., 2013b] Liu, J., Pasupat, P., Cyphers, S., and Glass, J. (2013b). Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390.

[Liu et al., 2023a] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

[Liu et al., 2019b] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Liu et al., 2023b] Liu, Y., Wei, S., Huang, H., Lai, Q., Li, M., and Guan, L. (2023b). Naming entity recognition of citrus pests and diseases based on the BERT-BiLSTM-CRF model. *Expert Systems with Applications*, 234:121103.

[Liu et al., 2020] Liu, Z., Luo, M., Yang, H., and Liu, X. (2020). Named entity recognition for the horticultural domain. In *Journal of Physics: Conference Series*, volume 1631, page 012016. IOP Publishing.

[Loshchilov and Hutter,] Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[Lothritz et al., 2020] Lothritz, C., Allix, K., Veiber, L., Bissyandé, T. F., and Klein, J. (2020). Evaluating pretrained transformer-based models on the task of fine-grained named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3750–3760.

[Ma et al., 2022] Ma, J., Ballesteros, M., Doss, S., Anubhai, R., Mallya, S., Al-Onaizan, Y., and Roth, D. (2022). Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971.

[Ma et al., 2020] Ma, T., Dou, Q., Jiang, P., and Liu, H. (2020). Named entity recognition based on semi-supervised ensemble learning with the improved tri-training algorithm. In *Proceedings of the 2020 8th International Conference on Information Technology: IoT and Smart City*, pages 13–18.

[Ma and Hovy, 2016] Ma, X. and Hovy, E. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*.

[Mackenzie et al., 2013] Mackenzie, J. S., Jeggo, M., Daszak, P., Richt, J. A., et al. (2013). *One Health: The human-animal-environment interfaces in emerging infectious diseases*, volume 366. Springer.

[Malarkodi et al., 2016] Malarkodi, C., Lex, E., and Devi, S. L. (2016). Named Entity Recognition for the Agricultural Domain. *Res. Comput. Sci.*, 117(1):121–132.

[Martinelli et al., 2015] Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., Villa, P., Stroppiana, D., Boschetti, M., Goulart, L. R., et al. (2015). Advanced methods of plant disease detection. A review. *Agronomy for sustainable development*, 35:1–25.

[Matúš, 1992] Matúš, F. (1992). On equivalence of Markov properties over undirected graphs. *Journal of Applied Probability*, 29(3):745–749.

[McCallum and Li, 2003] McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.

[McCann et al., 2017] McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.

[McIntosh, 2023] McIntosh, C. (2023). Cambridge English Thesaurus. Homepage at https://dictionary.cambridge.org/thesaurus. Last accessed 29 Nov 2023.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2011] Mikolov, T., Kombrink, S., Deoras, A., Burget, L., and Cernocky, J. (2011). RNNLM-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

[Morris et al., 2022] Morris, C. E., Géniaux, G., Nédellec, C., Sauvion, N., and Soubeyrand, S. (2022). One Health concepts and challenges for surveillance, forecasting, and mitigation of plant disease beyond the traditional scope of crop production. *Plant Pathology*, 71(1):86–97.

[Morwal et al., 2012] Morwal, S., Jahan, N., and Chopra, D. (2012). Named entity recognition using hidden Markov model (HMM). *International Journal on Natural Language Computing (IJNLC) Vol*, 1.

[Mulya and Khodra, 2023] Mulya, D. and Khodra, M. L. (2023). Biomedical event extraction using pre-trained SciBERT. *Journal of Intelligent Systems*, 32(1):20230021.

[Nédellec et al., 2024] Nédellec, C., Sauvion, C., Bossy, R., Borovikova, M., and Deléger, L. (2024). TaeC: A manually annotated text dataset for trait and phenotype extraction and entity linking in wheat breeding literature. *Plos one*, 19(6):e0305475.

[Ouyang et al., 2022] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

[Packer et al., 2010] Packer, T. L., Lutes, J. F., Stewart, A. P., Embley, D. W., Ringger, E. K., Seppi, K. D., and Jensen, L. S. (2010). Extracting person names from diverse and noisy OCR text. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 19–26.

[Panchendrarajan and Amaresan, 2018] Panchendrarajan, R. and Amaresan, A. (2018). Bidirectional LSTM-CRF for named entity recognition. In *Proceedings of the 32nd Pacific Asia conference on language, information and computation*.

[Patil et al., 2020] Patil, N., Patil, A., and Pawar, B. (2020). Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188.

[Peng et al., 2021] Peng, Q., Zheng, C., Cai, Y., Wang, T., Xie, H., and Li, Q. (2021). Unsupervised cross-domain named entity recognition using entity-aware adversarial training. *Neural Networks*, 138:68–77.

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

[Pergola et al., 2021] Pergola, G., Kochkina, E., Gui, L., Liakata, M., and He, Y. (2021). Boosting Low-Resource Biomedical QA via Entity-Aware Masking Strategies. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1977–1985.

[PESV, 2023] PESV (2023). PESV Homepage. https://plateforme-esv.fr. Last accessed 06 Feb 2023.

[Peters et al., 2017] Peters, M. E., Ammar, W., Bhagavatula, C., and Power, R. (2017). Semi-supervised sequence tagging with bidirectional language

models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[Pham et al., 2016] Pham, N.-Q., Kruszewski, G., and Boleda, G. (2016). Convolutional neural network language models. In *Proceedings of the 2016 conference on Empirical Methods in Natural Language Processing*, pages 1153–1162.

[Piantadosi, 2023] Piantadosi, S. (2023). Modern language models refute Chomsky's approach to language. *Lingbuzz Preprint, lingbuzz*, 7180.

[Pilehvar and Camacho-Collados, 2021] Pilehvar, M. T. and Camacho-Collados, J. (2021). *Contextualized Embeddings*, pages 69–96. Springer International Publishing, Cham.

[Popovski et al., 2019] Popovski, G., Kochev, S., Korousic-Seljak, B., and Eftimov, T. (2019). FoodIE: A Rule-based Named-entity Recognition Method for Food Information Extraction. *ICPRAM*, 12.

[Pushpalatha and Thanamani, 2019] Pushpalatha, M. and Thanamani, A. S. (2019). Rule Based kannada named entity recognition. *J. Crit. Rev*, 7:2020.

[Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., and Ghemawat, S. (2019). Language Models are Unsupervised Multitask Learners. In *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, pages 137–150.

[Raffel et al., 2020] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the Conf. on Empirical Methods in Natural Language*

Proc. and the 9th International Joint Conference on Natural Language Proc. (EMNLP-IJCNLP), pages 3982–3992.

[Ristaino et al., 2021] Ristaino, J. B., Anderson, P. K., Bebber, D. P., Brauman, K. A., Cunniffe, N. J., Fedoroff, N. V., Finegold, C., Garrett, K. A., Gilligan, C. A., Jones, C. M., et al. (2021). The persistent threat of emerging plant disease pandemics to global food security. *Proceedings of the National Academy of Sciences*, 118(23):e2022239118.

[Ritter et al., 2011] Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In Barzilay, R. and Johnson, M., editors, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[Roberts, 2001] Roberts, R. J. (2001). PubMed Central: The GenBank of the published literature.

[Rojas and Rojas, 1996] Rojas, R. and Rojas, R. (1996). The backpropagation algorithm. *Neural networks: a systematic introduction*, pages 149–182.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Salton, 1971] Salton, G. (1971). *The SMART retrieval system—experiments in automatic document processing*. Prentice-Hall, Inc.

[Sang and De Meulder, 2003] Sang, E. T. K. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Conf. on Natural Language Learning at HLT-NAACL*, pages 142–147.

[Sanh, 2019] Sanh, V. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of Thirty-third Conference on Neural Information Processing Systems (NIPS2019)*.

[Sari et al., 2010] Sari, Y., Hassan, M. F., and Zamin, N. (2010). Rule-based pattern extractor and named entity recognition: A hybrid approach. In *2010 International Symposium on Information Technology*, volume 2, pages 563–568. IEEE.

[Sarker, 2021] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160.

[Schoch et al., 2020] Schoch, C. L., Ciufo, S., Domrachev, M., Hotton, C. L., Kannan, S., Khovanskaya, R., Leipe, D., Mcveigh, R., O'Neill, K., Robbertse,

B., et al. (2020). NCBI Taxonomy: a comprehensive update on curation, resources and tools. *Database*, 2020:baaa062.

[Seideh et al., 2016] Seideh, M. A. F., Fehri, H., and Haddar, K. (2016). Named entity recognition from Arabic-French herbalism parallel corpora. In *Automatic Processing of Natural-Language Electronic Texts with NooJ: 9th International Conference, NooJ 2015, Minsk, Belarus, June 11-13, 2015, Revised Selected Papers 9*, pages 191–201. Springer.

[Sekine and Nobata, 2004] Sekine, S. and Nobata, C. (2004). Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *LREC*, pages 1977–1980. Lisbon, Portugal.

[Settles, 2004] Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLPBA/BioNLP)*, pages 107–110.

[Shen et al., 2023] Shen, Y., Song, K., Tan, X., Li, D., Lu, W., and Zhuang, Y. (2023). DiffusionNER: Boundary Diffusion for Named Entity Recognition. *arXiv preprint arXiv:2305.13298*.

[Soubeyrand et al., 2024] Soubeyrand, S., Estoup, A., Cruaud, A., Malembic-Maher, S., Meynard, C., Ravigné, V., Barbier, M., Barrès, B., Berthier, K., Boitard, S., et al. (2024). Building integrated plant health surveillance: a proactive research agenda for anticipating and mitigating disease and pest emergence. *CABI Agriculture and Bioscience*, 5(1):72.

[Sparck Jones, 1972] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

[Strauss et al., 2016] Strauss, B., Toma, B., Ritter, A., De Marneffe, M.-C., and Xu, W. (2016). Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144.

[Sun et al., 2021] Sun, L., Wang, J., Zhang, K., Su, Y., and Weng, F. (2021). RpBERT: a text-image relation propagation-based BERT model for multimodal NER. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13860–13868.

[Takeuchi and Collier, 2002] Takeuchi, K. and Collier, N. (2002). Use of support vector machines in extended named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

[Tan et al., 2021] Tan, Z., Shen, Y., Zhang, S., Lu, W., and Zhuang, Y. (2021). A sequence-to-set network for nested named entity recognition. *arXiv preprint arXiv:2105.08901*.

[Tedeschi et al., 2021] Tedeschi, S., Maiorca, V., Campolungo, N., Cecconi, F., and Navigli, R. (2021). WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533.

[van Rossum, 2022] van Rossum, G. (2022). Python programming language, v. 3.8.15. available at https://www.python.org/downloads/release/python-3815/. Last accessed 06 Feb 2023.

[van Rossum, 2024] van Rossum, G. (2024). Python programming language, v. 10.0.0. available at https://www.python.org/downloads/release/python-31014/. Last accessed 28 Jun 2024.

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

[Veena et al., 2023] Veena, G., Kanjirangat, V., and Gupta, D. (2023). AGRONER: An unsupervised agriculture named entity recognition using weighted distributional semantic model. *Expert Systems with Applications*, 229:120440.

[Verwimp et al.,] Verwimp, L., Pelemans, J., Wambacq, P., et al. Character-Word LSTM Language Models.

[Wallgrün et al., 2018] Wallgrün, J. O., Karimzadeh, M., MacEachren, A. M., and Pezanowski, S. (2018). GeoCorpora: building a corpus to test and train microblog geoparsers. *International Journal of Geographical Information Science*, 32(1):1–29.

[Wang et al., 2022] Wang, S., Li, X., Meng, Y., Zhang, T., Ouyang, R., Li, J., and Wang, G. (2022). k NN-NER: Named Entity Recognition with Nearest Neighbor Search. *arXiv preprint arXiv:2203.17103*.

[Wang et al., 2023a] Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., and Wang, G. (2023a). Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

[Wang et al., 2021] Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., and Tu, K. (2021). Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Comp. Ling. and the 11th International Joint Conference on Natural Language Proc.*, pages 2643–2660.

[Wang et al., 2023b] Wang, X., Zhou, W., Zu, C., Xia, H., Chen, T., Zhang, Y., Zheng, R., Ye, J., Zhang, Q., Gui, T., et al. (2023b). InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction. *arXiv e-prints*, pages arXiv–2304.

[Wang et al., 2019] Wang, Y., Li, Y., Zhu, Z., Xia, B., and Liu, Z. (2019). SC-NER: A sequence-to-sequence model with sentence classification for named entity recognition. In *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part I 23*, pages 198–209. Springer.

[Weischedel et al., 2013] Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., et al. (2013). Ontonotes release 5.0. *Linguistic Data Consortium, Philadelphia, PA*, 23:170.

[Wettig et al., 2022] Wettig, A., Gao, T., Zhong, Z., and Chen, D. (2022). Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*.

[White, 2020] White, J. (2020). PubMed 2.0. *Medical reference services quarterly*, 39(4):382–387.

[Wolf et al., 2020a] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020a). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

[Wolf et al., 2020b] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020b). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

[Xie et al., 2024] Xie, T., Zhang, J., Zhang, Y., Liang, Y., Li, Q., and Wang, H. (2024). Retrieval Augmented Instruction Tuning for Open NER with Large Language Models. *arXiv preprint arXiv:2406.17305*.

[Xu and Wang, 2021] Xu, A. and Wang, C. (2021). Ner based on feed-forward depth neural network. In *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, pages 510–516. IEEE.

[Yan et al., 2019] Yan, H., Deng, B., Li, X., and Qiu, X. (2019). Tener: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*.

[Yan and Li, 2021] Yan, L. and Li, S. (2021). Grape diseases and pests named entity recognition based on BiLSTM-CRF. In *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 4, pages 2121–2125. IEEE.

[Yang et al., 2024] Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., and Hu, X. (2024). Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

[Zhai et al., 2018] Zhai, Z., Nguyen, D. Q., and Verspoor, K. (2018). Comparing cnn and lstm character-level embeddings in bilstm-crf models for chemical and disease named entity recognition. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 38–43.

[Zhang et al., 2022] Zhang, L., Nie, X., Zhang, M., Gu, M., Geissen, V., Ritsema, C. J., Niu, D., and Zhang, H. (2022). Lexicon and attention-based named entity recognition for kiwifruit diseases and pests: A Deep learning approach. *Frontiers in Plant Science*, 13:1053449.

[Zhang et al., 2023a] Zhang, M., Yan, H., Zhou, Y., and Qiu, X. (2023a). Promptner: A prompting method for few-shot named entity recognition via k nearest neighbor search. *arXiv preprint arXiv:2305.12217*.

[Zhang and Elhadad, 2013] Zhang, S. and Elhadad, N. (2013). Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6):1088–1098.

[Zhang et al., 2015] Zhang, S., Zheng, D., Hu, X., and Yang, M. (2015). Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 73–78.

[Zhang et al., 2021] Zhang, X., Xu, G., Sun, Y., Zhang, M., and Xie, P. (2021). Crowdsourcing Learning as Domain Adaptation: A Case Study on Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5558–5570.

[Zhang et al., 2023b] Zhang, Y., Pu, P., Huang, L., Qian, B., and Liu, Y. (2023b). Chinese named entity recognition of apple diseases and pests based on iterative dilated convolution. In *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 1810–1815. IEEE.

[Zhao et al., 2022] Zhao, P., Wang, W., Liu, H., and Han, M. (2022). Recognition of the agricultural named entities with multifeature fusion based on albert. *IEEE Access*, 10:98936–98943.

[Zheng et al., 2021] Zheng, H., Yu, H., Hao, Y., Wu, Y., and Li, S. (2021). Distantly supervised named entity recognition with Spy-PU algorithm. In *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 56–63. IEEE.

[Zheng et al., 2022] Zheng, J., Chen, H., and Ma, Q. (2022). Cross-domain named entity recognition via graph matching. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2670–2680.

[Zhou and Su, 2002] Zhou, G. and Su, J. (2002). Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 473–480.

[Zhou et al., 2023] Zhou, W., Zhang, S., Gu, Y., Chen, M., and Poon, H. (2023). UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition. In *The Twelfth International Conference on Learning Representations*.

[Zhou, 2018] Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53.

[Zhu et al., 2020] Zhu, H., He, C., Fang, Y., and Xiao, W. (2020). Fine grained named entity recognition via seq2seq framework. *IEEE Access*, 8:53953–53961.