Ca' Foscari
University
of Venice

Master's Degree

in

Data Analytics for Business
and Society

Final Thesis

Exploring Cross-Lingual Named Entity Recognition: A Study of the ConNER Model for the Italian Language

Supervisor
Ch. Prof. Andrea Albarelli

Assistant supervisor
Ch. Prof. Raffaele Pesenti

Graduand
Francesca Ferraresso
866698

Academic Year
2022 / 2023

Abstract

This dissertation provides a comprehensive overview of Named Entity Recognition (NER) in Natural Language Processing (NLP). It focuses on cross-lingual NER models, exploring how they leverage shared knowledge among languages to enhance their performance. We investigate the advantages and limitations of cross-lingual NER. An important aspect is the analysis of ConNER, a state-of-the-art cross-lingual NER model, with a focus on its performance in the Italian language. Our empirical study employs a modified MultiNERD dataset covering English, German, French and Spanish, shedding light on ConNER's adaptability to other languages.

Index

Introduction

Natural Language Processing (NLP) is a subfield of Computer Science concerned with making human natural language understandable to computers. In the past years, we have seen its rapid development thanks to an increase in the amount of data available and the progress made in the field. Named Entity Recognition (NER), a subtask of NLP whose goal is to extract and classify entities from textual data, is becoming more and more studied because of its uses in various NLP tasks.

Our objective is to provide a well-rounded explanation of NER's main concepts and principles, its historical background and the most recent and important state-of-the-art NER techniques.

Another contribution of this work is the exploration of methods for cross-lingual NER, a specific subtask whose goal is to bridge the linguistic gap between languages. We will give a description of its models, mechanisms and techniques.

Moreover, we will provide a description of the advantages and disadvantages of cross-lingual NER techniques. We will discuss topics such as the benefits of reduced data annotation and improved generalisation along with the challenges related to language variations and the scarcity of linguistic resources.

Lastly, we will conduct an original analysis of ConNER, a state-of-the-art NER model. It will be trained on a version of the MultiNERD dataset and we will assess the model performance in transferring NER capabilities from English, French, Spanish and German languages to Italian.

Our ultimate goal is to contribute to the research and explore new possibilities to improve NER systems using cross-lingual approaches.

Chapter I

Named Entity Recognition: Concepts, Foundations and Techniques

1.1 Concept and Significance of NER

Named Entity Recognition allows us to extract information and analyse texts related to different contexts and fields. This process takes care of identifying and categorising named entities, which could be an individual, a location, a date, an organisation and more.

We should keep in mind that NER plays a big part in making easier for computers to understand human language. Starting from large amounts of data, we can access new information and understand individual relationships.

The concept of NER is closely linked to understanding the context at hand as well as linguistic features, like grammatical structures and syntactic clues like a human being would (Li et al., 2017).

For example, in the sentence "Apple unveiled its product", we can understand that "Apple" refers to an organisation based on the sentence's context and because it starts with a capital letter, while the word "product" is a term. This example highlights the ambiguity of handling entities and recognising them according to the context since "Apple" could have referred both to an organisation or a fruit.

1.2 Main applications of NER

This section explores some of the main applications of NER.

1.2.1 Information Extraction

"Information extraction structures extracted text according to entity recognition and entity relationships, which, then, feed into downstream search and query-based activities." (Olivetti et al., 2020, p. 4). Information extraction systems receive natural language text in input and generate structured information based on specific criteria that are relevant to a specific application. This application is time-saving and cost-effective if we consider the amount of data and the variety of contexts at one's disposal. It can be a great solution in many areas, for example, text extraction in the medical field (Olivetti et al., 2020).

1.2.2 Information Retrieval (IR)

Nowadays, finding information in a fast and practical way is essential to satisfy a wide variety of users. So, the increasing number of materials and websites on the Internet can prevent users from finding the information they need (Kobayashi et al., 2000). That is why NER has the specific duty to "[..] search a collection of natural language documents to retrieve exactly the set of documents that pertain to a user's question" (Voorhees, 1999). A great example of this application is search engines such as Google.

1.2.3 Supporting Knowledge Graph Construction

A graph matches the definition of a set of nodes connected by edges (Majeed and Rauf, 2020, p. 2). A knowledge graph (KG) is structured through a set of triples. "A triple is a 3-tuple (h, r, t) where h represents a head entity, t represents a tail entity, and r expresses a relationship between the two entities." (Kejriwal, 2019). A KG is a way of transferring knowledge, whether domain-specific or generic, in a way computers can understand. Thanks to NER, it is possible to facilitate the text-mining process to make the construction of KGs faster and more efficient.

1.2.4 Question Answering Systems

Question Answering Systems (QAS) has the purpose of putting together an answer starting from a natural language question. To get our answer, it is necessary to implement a series of steps. The question has to be processed through analysis and reformulation of the main keywords, then the information pertinent to the topic is filtered and ordered, and finally, the answer is extracted and validated (Allam et al., 2012). Using NER in QAS is helpful since many fact-based answers to questions are related to named entities that can be detected (Mollà et al., 2006).

1.2.5 Machine Translation Systems

These days the widespread use of the Internet connects individuals leading to the increasing need to understand as new content becomes available. Machine Translation (MT) is becoming more and more popular for this reason. To build accurate MT systems, there must be a database selection and keyword search followed by a qualitative analysis process where connections across different languages are formed and reviewed (Rivera-Trigueros, 2021, p. 7). A famous example of MT systems is Google Translate. NER can improve MT systems by recognising names, leading to a more accurate translation. Moreover, it can help give information about the context of the text as well as reduce ambiguity (Shah et al., 2010)

1.2.6 Entity Linking

NER is a prerequisite for Entity Linking (EL), where ambiguous textual mentions are associated with specific named entities. EL is achieving more accurate results thanks to the application of different techniques like enhanced entity representation, NER-constrained decoding strategy or a combination of both (Tedeschi et al., 2021).

1.3 The Historical Development and Evolution of NER

The evolution of NER throughout the years was possible as a result of previous developments and discoveries in the NLP field.

Late 1980s and Early 1990s — NER models began to rise in popularity with the introduction of statistical machine learning techniques in the 1990s. This led to their applications in domain-specific areas to extract named entities and recognise patterns of different kinds. Hidden Markov Models (HMMs) were previously applied in computational biology (Churchill, 1989) and there were now different kinds of HMMs, all of which were based on HMM theory (Eddy, 1998). A notable example is the development of term frequency–inverse document frequency (tf–idf) by Robetson and Spärck Jones. Tf–idf measures the importance of a term within a document. It is now the foundation for IR systems employed in search engines; a study in 2015 showed that tf-idf was one of the mainly used weighing schemes (Beel et al., 2016)

The 2000s and early 2010s — In 2002 and 2003 respectively, CoNLL 2002 (Tjong Kim Sang, 2002) and CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) led to the creation of two benchmark datasets that are still widely used today. Systems began to include part-of-speech tagging, syntactic parsing and gazetteers to obtain better results. Researchers began to investigate the use of semi-supervised and unsupervised NER methods to deal with the drawbacks of having insufficient labelled data. Abney (2004) and Haffari & Sarkar (2007) implemented semi-supervised learning by giving a thorough explanation and application of the Yarowsky algorithm. The Yarowsky algorithm is a semi-supervised learning approach that aims to improve the accuracy of NER systems by iteratively updating and refining their training data.

The mid and late 2010s — The introduction of word embeddings like Word2Vec, GloVe and FastText greatly improved how words and context are represented in NER models. These embeddings try to encode the words' meaning into a machine-interpretable representation, thereby bridging the gap between human comprehension of language and its machine representation.

The Word2Vec algorithm, which was created by Tomas Mikolov et al., in 2013, is a method that represents words as vectors (word embeddings) in a vector space. It assigns each word in each text corpus to a position within a space vector, where words that have meanings are located close to each other. Word2Vec is based on a feed-forward neural network consisting of two layers; the Projection Layer and the Fully Connected Layer.

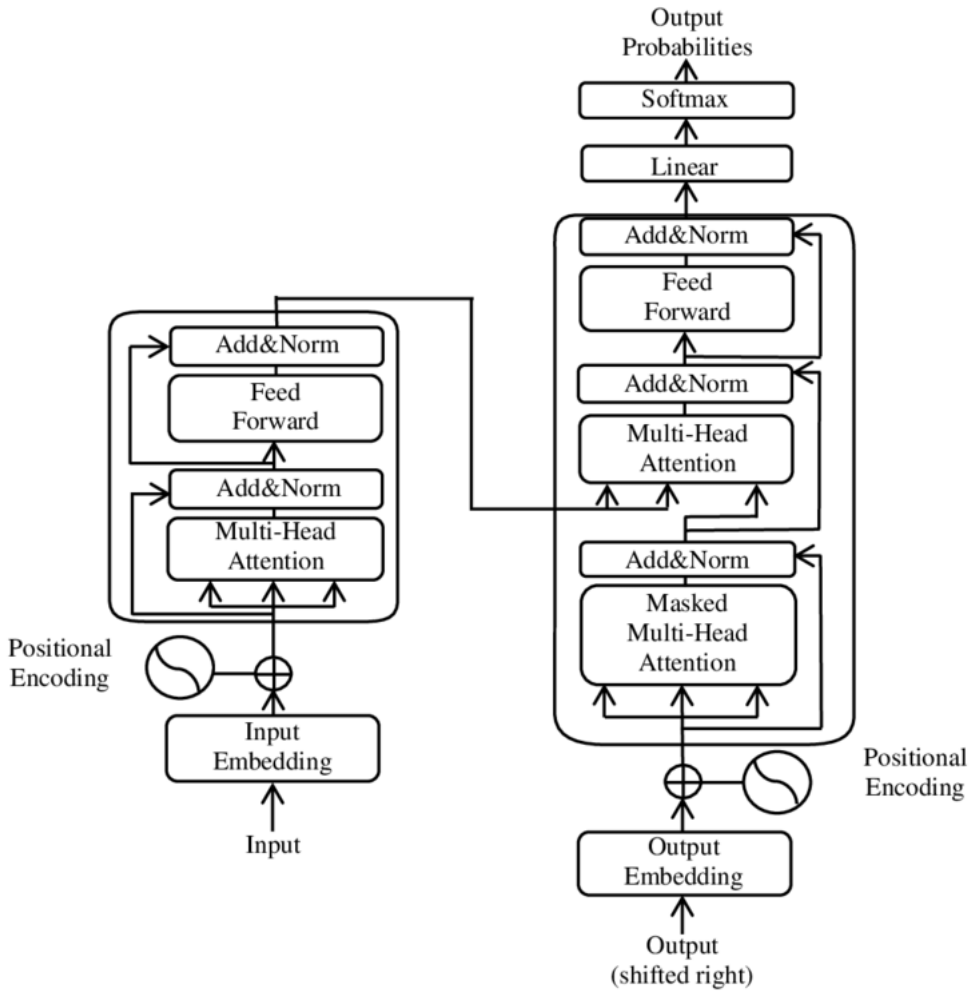
The Projection Layer captures the relationships between words, while the Connected Layer is trained to predict the context for a given target word.

Furthermore, new techniques such as zero-shot and few-shot NER were introduced. In zero-shot learning (Palatucci et al, 2009), during the testing phase, a learner is presented with samples from classes that were not encountered during the training phase. The learner's task is to accurately predict the class to which these samples belong. Few-shot learning uses only a few labelled samples per class.

In 2017, the paper “Attention is All You Need” by Vaswani et al. introduced Transformers. A Transformer is a type of deep-learning model that processes sequential data and focuses on identifying the relationships between elements (Vaswani et al., 2017).

It operates by processing input text in a structured manner. Initially, the input text is divided into tokens using a byte pair encoding tokeniser. These tokens are then converted into vectors through a word embedding table. Notably, positional information is added to these embeddings to ensure the model understands the order of tokens within the sequence.

Fig. 1: Transformer model architecture



Source: Vaswani et al., 2017

The Transformer architecture (Fig. 1) adopts an encoder/decoder framework. In the encoder, multiple layers iteratively process the input tokens. Each layer's primary function is to create contextualised token representations, achieved through self-attention mechanisms. Conversely, the decoder also comprises layers but processes both the encoder's output and its own generated tokens. Each decoder layer incorporates two types of attention: cross-attention, which interprets the encoder's output, and self-attention, which mixes information among the decoder's input tokens during inference. Additionally, both encoder and decoder layers encompass feed-forward neural networks, residual connections, and layer normalisation for effective information flow.

At the heart of the Transformer's functionality is the scaled dot-product attention mechanism. This mechanism employs query, key, and value weight matrices for each token. It computes attention weights by taking the dot product of query and key vectors, followed by normalisation through softmax. The final output is a weighted sum of value vectors across all tokens.

Furthermore, the Transformer leverages multi-head attention, with each layer containing multiple attention heads. These heads enable the model to discern various aspects of token relevance and facilitate a broader understanding of token relationships. To ensure contextually appropriate attention, masking may be applied. For example, during autoregressive text generation, future tokens are masked to prevent reverse information flow. Incorporating positional information is essential for the Transformer to understand token order, and this is achieved through positional encoding, a fixed-size vector representation.

Among transformer-based models, we can find the Bidirectional Encoder Representations from Transformers model, or BERT, developed by Devlin et al. in 2018. BERT is a model that makes use of the Transformer architecture, which relies on an attention mechanism. Unlike the Transformer architecture that has both encoder and decoder components, BERT is solely based on the encoder mechanism. Additionally, BERT utilizes masked language models to allow for trained deep bidirectional representations.

1.4 The Main Challenges and Limitations of NER

One of the most fundamental challenges in NER is the ambiguity and context dependency of the named entities. Indeed, entities often have multiple meanings based on their surrounding context. For example, the word "Crane" can refer to both a bird species and a construction machine. Creating NER models that can properly use contextual information is very important if we want to find effective solutions to this problem.

Another critical challenge in NER is the lack of annotated data. (i.e., large datasets where entities are properly extracted and labelled by experts). The performance of many NER models depends on the availability of annotated training data (Smith et al., 2021). However, obtaining large-scale labelled datasets in various domains and languages can be expensive and time-consuming. One of the most popular approaches to overcome this limitation is transfer learning, in which knowledge is transferred between tasks to improve performance. For instance, researchers may fine-tune–pre-trained models on limited NER labelled data, such as to make NER more adaptable and transferable to new tasks.

Named entities can be found in different forms, especially in informal or colloquial texts. For example, the acronym "USA" and the words "United States of America" both refer to the same entity, despite their syntactic differences.

The issue of out-of-distribution entities is also important (Fort et al., 2021). This problem occurs when entities are not observed during training, which is frequently the case for domain-specific terms, causing NER models to struggle with unseen instances. Furthermore, named entities may overlap or nest with each other, resulting in additional issues. For example, the outer entity “the Oakland Zoo” contains an inner entity, “Oakland” (Wang et al, 2022).

1.5 Different Types of Named Entities

Named entities are items or components with proper names, such as individuals, groups, places, and dates. NER is essential for text interpretation and information extraction in various NLP applications. In this in-depth investigation, we look at the many kinds of named entities frequently found in NER and discuss the state of the research in this field.

Names of People (PER): Person names [1] include all terms that may be used to characterise the name of a person, including titles (such as Mr., Mrs., and Dr.), first names (i.e. John) and last names (i.e. Smith), and complete names (i.e. John Smith).

These entities are used in sentiment analysis, social network analysis and much more. (Medhat et al., 2014)

[1] Sarah Brown (PER) is our new neighbour.

Organisation Names (ORG): The names of businesses, institutions, governmental agencies, and other corporate entities are represented by organisation names [2]. Recognising and categorising organisational names when performing tasks like market analysis, business intelligence, and news monitoring regarding corporations is essential (Chilet et al., 2016).

[2] The United Nations (ORG) held a conference.

Geographical Names (LOC): Location names [3] include designations for geographic locations like cities, nations, states, regions, landmarks, and addresses. Geographical entities are needed for geospatial analysis, mapping, and location-based services (Leidner et al., 2003)

[3] The United States (LOC) is a large country.

Date and Time Expressions (DATE): Date and time expressions [4] may include any specific references to dates, days of the week, months, years, or lengths of time. Precise recognition of date and time entities is important for tasks involving event extraction, temporal reasoning, and timeline generation (Mirza, 2016).

[4] The event will last for three days (DATE).

Numerical Entities (NUM): Numerical entities [5] include numerical expressions such as cardinal and ordinal numbers, percentages, and monetary values. Recognising numerical entities is necessary for financial analysis, statistical reporting, and the extraction of numerical facts (Loukas et al., 2022)

[5] 80% (NUM) of customers recommend it.

Product Names (PROD): Product names [6] consist of the names of commercial products, brands, and trademarks. Product name identification is critical for tasks such as sentiment analysis of product reviews and product launch monitoring (Vinodhini et al., 2012).

[6] I bought the new iPhone 13 (PROD).

Miscellaneous Entities (MISC): Miscellaneous entities [7] encompass various named entities that do not fall into the above categories. This category may include specialised terms, domain-specific entities, and other unique identifiers (Vinodhini et al., 2012).

[7] DNA (MISC) carries genetic information.

1.6 Approaches and Techniques in NER

A great number of NER models and approaches have been developed to overcome issues and they could be applied in many areas. In this section, we will give a review of rule-based, learning, deep learning, ensemble, and knowledge-based approaches. We will highlight their characteristics and consider the influence their introduction had on this discipline.

1.6.1 Rule-based Approaches

Rule-based (RB) approaches, or Knowledge-based approaches, function thanks to specific rules. These rules are usually created by humans and have changed as a result of real-life experiences (Thanaki, 2019).

1.6.1.1 Process

There is a series of steps to follow when implementing the framework of RB approaches.

First of all, if the rules to be applied do not exist yet, they can be created. They are domain-specific according to the desired task. Since RB approaches analyse textual data, they can refer to grammar, syntax or lexicon. The rules are then applied to the input data. As the system analyzes the text it will carefully evaluate the results of these rules. Doing it will extract information from the text. It is important to mention that these rules are updated based on feedback received from previous inputs.

1.6.1.2 Categories

Rule-based approaches can be categorised according to different aspects (Table 1). These include the number of inputs and outputs, their types, as well as the type of structure, logic, and environment.

Table 1: RB approaches categorisation

Categorization Criteria	**RB Approach Types**
Number of Inputs/Outputs	Single-Input-Single-Output
	Multiple-Input-Single-Output
	Single-Input-Multiple-Output
	Multiple-Input-Multiple-Output
Type of Inputs/Outputs	Rule-Based Classification Systems
	Rule-Based Regression Systems
	Rule-Based Association Systems
System Structure	Networked Rule-Based Systems

	Listed Rule-Based Systems
	Treed Rule-Based Systems
Type of Logic Used	Fuzzy Rule-Based Systems
	Probabilistic Rule-Based Systems
	Deterministic Rule-Based Systems

Inputs and outputs for RB approaches can be single or multiple. So, approaches can be divided into four types: single-input-single-output, multiple-input-single-output, single-input-multiple-output, and multiple-input-multiple-output (Liu et al., 2014).

Another type of categorisation, based on the number and type of inputs and outputs, sees rule-based systems divided into three categories: rule-based classification systems, rule-based regression systems, and rule-based association systems. (Liu et al., 2014)

In terms of their structure, these systems can be divided into three groups: networked rule-based systems, listed rule-based systems, and treed rule-based systems. (Liu et al., 2014)

Lastly, any rule-based system will be constructed using specific types of logic, such as boolean logic, fuzzy logic, and probabilistic logic. Consequently, they may also be divided into three categories: fuzzy rule-based systems, probabilistic rule-based systems, and deterministic rule-based systems. (Liu et al., 2014)

1.6.1.3 Advantages and Limitations

RB approaches are transparent, and they can be easily interpreted since the rules applied are defined and understandable**.** They can be made domain-specific and customised easily since the rules are created by humans. They do not require a large amount of labelled data when trained.

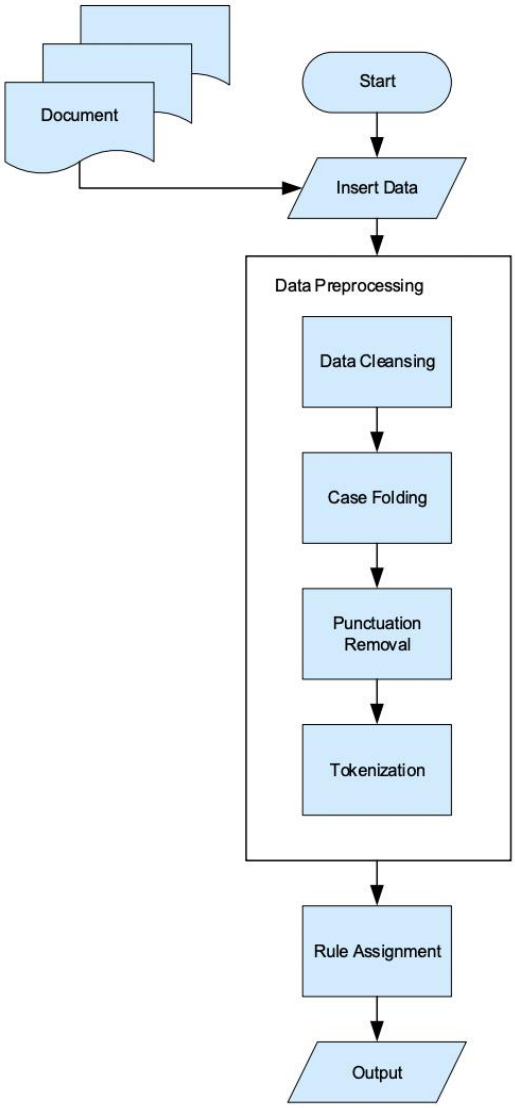
However, creating and refining rules takes a great effort. These approaches may also not perform well when the context is ambiguous and may be subjected to human errors.

1.6.1.4 Example of Rule-based Approach

A practical example of rule-based approach applied to NER is brought by Wahyuni et al., 2021 to recognise time expressions present in Balinese text documents. This is particularly useful since time expressions usually reference events, facts or information. The rules used have been created by combining contextual, morphological, and part-of-speech knowledge.

The process, as shown in Fig. 2, includes the identification of date and time expressions, and then an output is generated according to the rules applied. After, the data is cleaned and split into smaller expressions called tokens. The time expression entities are extracted and given in output.

Fig. 2: NER development process



Source: Wahyuni et al., 2021

1.6.2 Learning Approaches

Machine learning (ML) is a subfield of Artificial Intelligence (AI) which includes all those algorithms built with the goal of learning to recognise complicated patterns, label sequences and make decisions based on data.

In the context of ML, features are measurable properties of the data. In computational linguistics, these refer to the characteristics of text objects (words, sentences, etc.). Various factors, including language, domains and the qualitative and quantitative characteristics of training data influence the usefulness of the features. As a result, feature selection is usually task-specific, and it can frequently result in varied performance in NER systems.

Broadly speaking, the aim of learning algorithms is to automatically detect patterns from the data. This should allow them to gather comparable information in unseen data by learning semantic, syntactic and contextual aspects of the training data.

Based on the training data and how it is used, there are three main categorisations of learning algorithms: supervised, semi-supervised, and unsupervised.

Supervised learning uses labelled data to construct its models, *i.e.*, data tagged with one or more labels that are used to guide the training process. Semi-supervised learning, on the other hand, seeks to mix labelled data (usually in small amounts) with meaningful information from unlabelled data to improve learning. Lastly, unsupervised learning approaches include all those models that attempt to learn from unlabelled data.

1.6.2.1 Supervised Learning

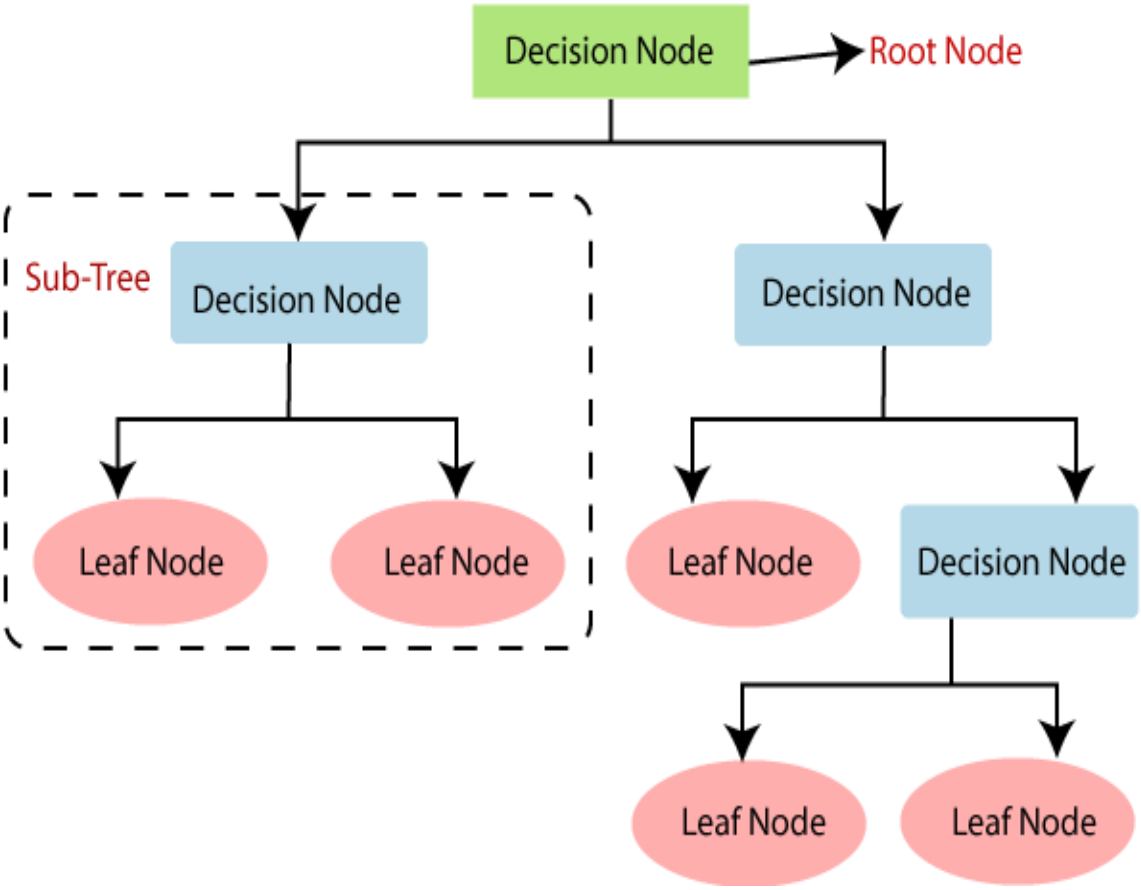
A supervised learning system uses training data and associated properties as input to generate an extraction model, subsequently used to recognise similar objects in unseen data.

Examples of supervised learning approaches are Decision Trees, Naive Bayes, Support Vector Machines, Linear Regression and Neural Networks.

- **Decision Tree Learning** is a supervised learning approach used in statistics, data mining and machine learning. The objective of this approach is to create a model that predicts the value of a target variable based on several input variables.

A decision tree starts at the root node, which contains all the information of the dataset. From there, it selects a characteristic to analyse, known as a classification. Once it identifies the attribute, the data is divided into groups called child nodes or branches based on their values. This process continues for each branch selecting attributes for each subgroup. When certain conditions are met, such as tree depth or sample size, the algorithm creates endpoint nodes referred to as leaf nodes. These leaf nodes represent categories or predicted values (as shown in Fig. 3)

Fig. 3: Decision Tree structure



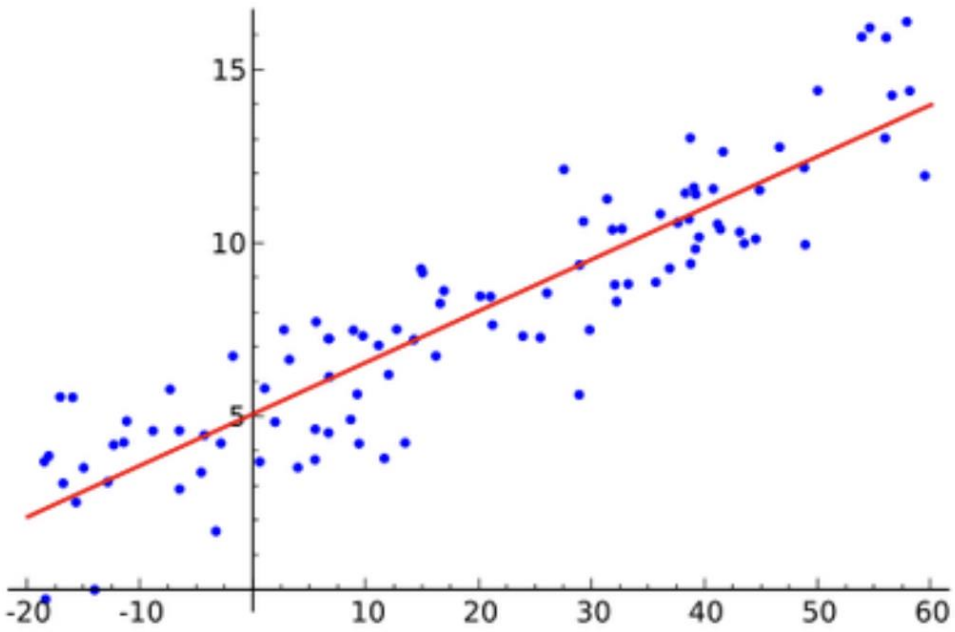
- **Linear regression** establishes connections between variables. It involves modelling the relationship between a dependent variable (also known as the label or target) denoted as 'y' and one or more explanatory variables (also referred to as independent variables) represented by 'X' using a linear function. In regression analysis, the aim is to predict a target variable while classification focuses on

predicting labels from a set. For regression models that involve a combination of input variables they can be represented in the following form:

$$y = \beta 0 + \beta 1\, x1 + \beta 2\, x2 + \ldots + e$$

In Figure 4, the model represented by the red line is created using a set of training data points (depicted in blue). Each blue point has a known label on the y-axis. The objective is to make the model closely match these points by minimizing a chosen loss function.

Fig. 4: Linear Regression representation

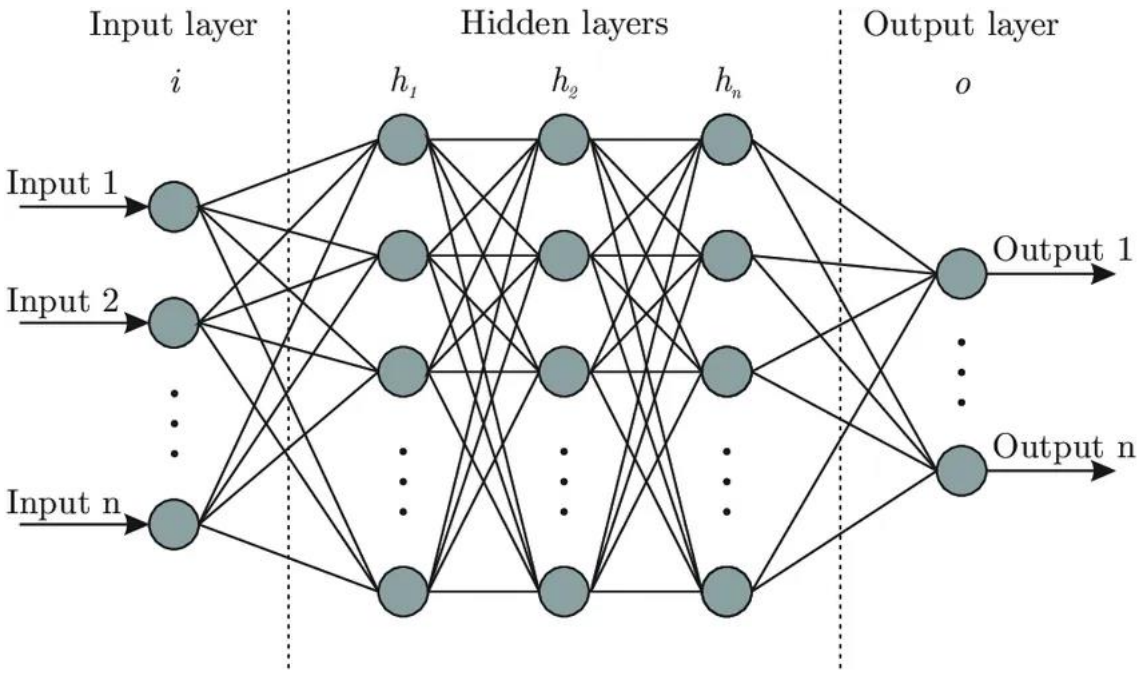


Source: Nasteski, 2017

- **Neural Networks (NNs)** consist of three components: input, output and hidden layers (Fig. 5). These artificial neural networks (ANNs) are machine learning models that aim to replicate the brain's processing capabilities. They operate by passing data through layers of interconnected neurons.

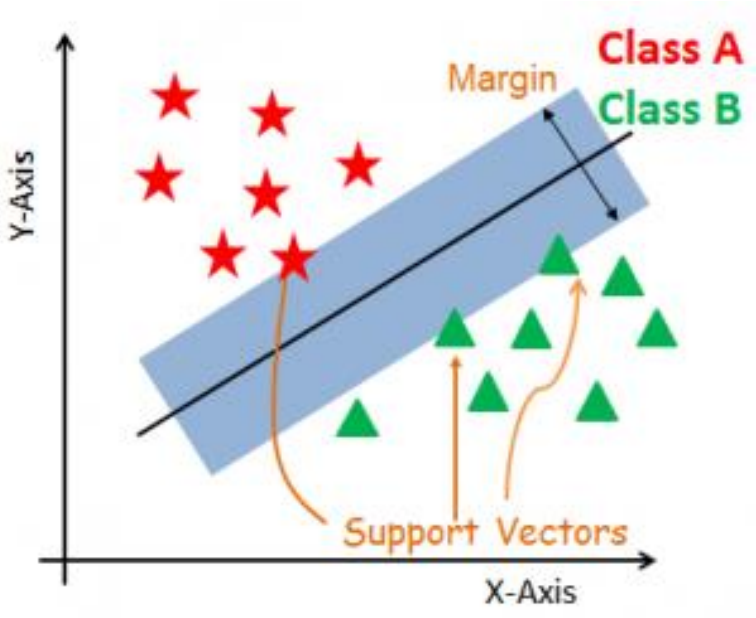
Firstly, there is the input layer, which initially introduces data into the model for training and learning purposes. Secondly, weight parameters are utilised to organise variables by assigning importance and measuring their impact on the model's predictions. The transfer function serves as a component for aggregating and combining all input information into a single output variable. This step plays a role in information processing. Lastly, we have the activation function acting as a decision-maker. It determines whether a specific neuron should activate, based on its perceived relevance, to the prediction process.

Fig. 5: Neural Network architecture



- **Support Vector Machines (SVMs)** are models developed by Vapnik et al. between 1992 and 1997. They find the optimal line or hyperplane that maximises the distance between the data points of different classes. This hyperplane is like a decision boundary that helps classify unseen data points into one of the predefined categories (Fig. 6)

Fig. 6: SVMs architecture



- **Naive Bayes (NB) classifiers** use the probability of a given data point belonging to a class to make predictions. It is called "naive" because it assumes that the features are conditionally independent meaning that each feature independently contributes to the probability of the data points class. By multiplying these probabilities Naive Bayes calculates the probability of the data point falling into a specific class. Finally, it predicts the class with the probability. This approach works well for tasks, like text classification, spam filtering and sentiment analysis because it simplifies computation while still offering results (Nasteski, 2017).

1.6.2.2 Semi-supervised Learning

Semi-supervised learning aims to enhance learning performance by leveraging both labelled and unlabeled data without the need for human intervention (Zhou, 2018).

Bootstrapping (or self-training) is a notable kind of semi-supervised learning in NER. The system is first trained on a limited set of samples and then used to tag unlabelled data. The generated annotations are then utilised to supplement the initial training dataset

and subsequently used to retrain the system. The method is repeated numerous times to refine the learning judgments and annotate the results (Thelen et al., 2002)

Weakly supervised learning can be considered a type of semi-supervised learning. It refers to a range of research areas that aim to build models by using supervision during the learning process.

There are three notable forms of weak supervision: incomplete supervision, where only a portion of the training data is provided with labels; inexact supervision, where the training data is labelled in a more general manner; and inaccurate supervision, where the provided labels may not always be entirely accurate (Zhou, 2018)

1.6.2.3 Unsupervised Learning

Unsupervised learning approaches process unlabelled data to identify patterns and hidden structures within the dataset.

Examples of supervised learning approaches are K-means Clustering, Principal Component Analysis (PCA) and Factor Analysis.

- **K-means clustering** is a technique that groups data into clusters based on their similarities. In this method, each data point is assigned to the centre of a cluster collection. The cluster centres are then updated by averaging the data points assigned to each cluster. This process continues until the centres become relatively stable or a predetermined number of iterations are completed. As a result, we obtain clusters of data points (Eckhardt et al., 2022).

- **Principal Component Analysis (PCA)** is a technique that helps simplify data by preserving its important information while reducing its dimensionality. It does this by transforming the features into a set of variables called principal components (PCs), which are combinations of the original features. The goal is to capture as much of the data variability as possible through these components. To use PCA effectively it is important to preprocess the input data by

standardizing and scaling. PCA finds applications, in fields including image processing, genetics and finance to simplify data representations and uncover underlying patterns (Eckhardt et al., 2022).

- **Factor analysis** is an alternative dimensionality reduction tool that extracts the common variance in the dataset in the form of unobserved or latent factors. The user must pre-specify the optimal number of factors and must ensure that there is some degree of correlation in the dataset (Eckhardt et al., 2022).

1.6.3 Transfer Learning Approaches

"Transfer learning refers to a set of methods that extend this approach by leveraging data from additional domains or tasks to train a model with better generalisation properties." (Ruder et al., 2019)

The goal of transfer learning is to use the knowledge learnt from one task to improve the performance of another. Transfer learning is useful in that it allows us to effectively handle data scarcity, improve model convergence, and achieve state-of-the-art performance in numerous scenarios.

These types of approaches present both advantages and limitations. As mentioned, a positive aspect of transfer learning is that it enables models to perform well even with small amounts of labelled data for a given task. This is because the pre-trained model already captures a significant amount of general information from the vast amount of data on which it has been trained. In addition, training a model from scratch on a given task can be computationally costly and time-demanding; in this case, as the starting model has already learnt low-level characteristics, transfer learning can significantly speed up the training process.

However, since pre-trained models are trained from large amounts of diverse datasets, they are commonly very general in terms of domain and thus will not be focused on any topic specifically. The success of a transfer learning approach might be determined by how closely the pre-trained task coincides with the target task (Mehra et al., 2023). This

is because pre-trained characteristics may not be immediately relevant, resulting in inferior performance. Moreover, since pre-trained models are highly complex, it might be difficult to explain how they arrive at certain predictions. Lastly, while transfer learning decreases the requirement for labelled data, it still requires task-specific labelled data for fine-tuning. As said before, obtaining high-quality labelled data may be both costly and time-consuming.

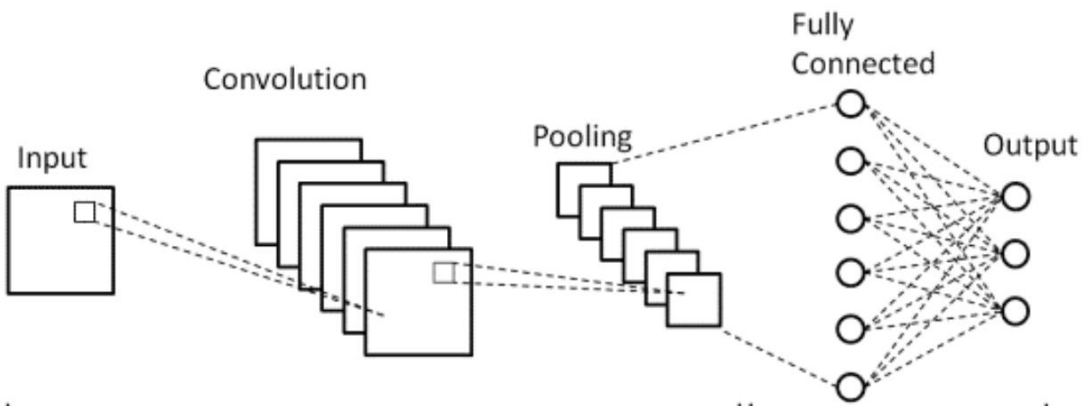
1.6.4 Deep Learning Approaches

Deep learning (DL) approaches fall under the umbrella of ML and they are based on neural networks. It is defined as "deep" learning because of its use of multiple layers in the network.

Listed below there are three of the most common Deep Neural Network (DNN) architectures:

- A **Convolutional Neural Network (CNN)** has a structured architecture for processing grid-like data like images (Shrestha et al., 2019). As shown in Fig. 7, it begins with convolutional layers that detect patterns, followed by pooling layers to downsample the features. Next, fully connected layers make predictions with non-linear activation functions like ReLU (rectified linear unit). The network learns through backpropagation, adjusting weights and biases.

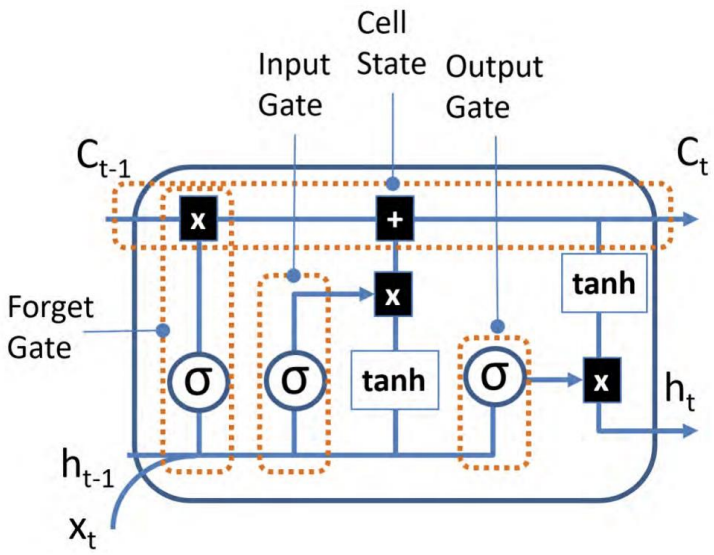
Fig.7: CNN architecture



- An **autoencoder** is a type of neural network architecture that focuses on dimensionality reduction. It incorporates two key components: an encoder and a decoder. The encoder takes input data and compresses it into a reduced representation known as the latent space. Conversely, the decoder's role is to reconstruct the initial input data based on this compressed representation. During training, the network's primary objective is to minimise the reconstruction error, ensuring that the essential information is retained in the compressed representation. Autoencoders find practical use in tasks like data compression, denoising, and feature extraction (Shrestha et al., 2019).

- The **Long Short-Term Memory (LSTM)** is an implementation of recurrent neural network (RNN) architecture (Hochreiter et al., 1997). It has the ability to handle sequential data thanks to its architecture made up of memory cells and gating mechanisms. These include an input gate, a forget gate, and an output gate, which control the information flow as show in Fig. 8. LSTMs are a useful tool in tasks related to natural language processing, speech recognition, and time series prediction (Shrestha et al., 2019).

Fig.8 : LSTM architecture



Source: Shrestha et al., 2019

1.6.5 Ensemble Approaches

Ensemble methods combine multiple independent models in order to obtain more accurate predictions compared to the single models. They use various strategies to aggregate the predictions of individual models, including majority voting, weighted averaging, and stacking (Kunapuli, 2023).

The ensemble process typically calls for the training of numerous models on the same or distinct datasets and then integrates their outputs to make final predictions.

These techniques are flexible and can be applied to rule-based, statistical, or machine-learning-based NER models, allowing for a comprehensive approach to entity recognition (Sagi et al., 2018).

Two popular examples of ensemble methods are bootstrap aggregation (or bagging) and boosting:

- **Bootstrap aggregation** (Breiman, 1996) uses multiple copies of the same model trained on different samples. The result corresponds to the average of the predictions of multiple models. This can reduce variance and give a more robust prediction (Sutton, 2005).

- **Boosting** uses a sequence of weak models that are trained subsequentially and incorporates weights. In each step, the samples used are all from different populations and the incorrect prediction from a step receives an increased weight in the following steps (Sutton, 2005).

Their key advantage is their ability to produce more reliable and accurate predictions by considering diverse perspectives and compensating for the shortcomings of individual models. An obvious disadvantage, however, is the fact that a variety of models have to be trained, which is why ensemble models are typically composed of weak components to ease the learning process.

1.6.6 Knowledge-based Approaches

"Knowledge-based methods [..] are systems that, in addition to using linguistic knowledge, also rely on explicitly formulated domain or world knowledge to solve typical problems in Natural Language Processing such as ambiguity resolution and inferencing." (Nirenburg and Mahesh, 1997)

Most knowledge-based systems need to acquire knowledge of the domain of study to be able to extract and manipulate textual data. Knowledge-based approaches need to incorporate and apply such knowledge to solve problems such as ambiguity resolution (Mahesh and Nirenburg, 1997)

Knowledge-based solutions have considerably impacted the area of NLP, giving a viable alternative to approaches based on linguistic information. Thanks to these systems, numerous AI models can now be grounded in real-world input and output as natural languages.

Chapter II

Cross-Lingual Named Entity Recognition

2.1 Concept

As we have illustrated in the previous chapter, NER is indispensable for carrying out a great variety of tasks. Nonetheless, NER is carried out successfully if the languages analysed have a considerable amount of annotated data available, otherwise, the whole process becomes challenging.

In cases such as this, we introduce the concept of cross-lingual Named Entity Recognition, an NLP subtask that deals with identifying and classifying named entities across different languages.

2.2 Advantages and Limitations

This kind of approach has several advantages. Firstly, they promote language diversity since they enable the analysis of multiple languages, even though they present a limited number of labelled data. Cross-lingual NER models are widely used to solve problems related to low-resource languages; in fact, they aid the transfer of knowledge from high-resource languages, such as English, to low-resource ones, such as Yoruba and Burmese.

Furthermore, cross-lingual NER models assure consistency across languages as well as allow knowledge sharing, where models are given the ability to capture linguistic and cultural variations across languages, also leading to improved accuracy and robustness of the model.

Lastly, these approaches include reduced annotation costs through transfer learning, allowing the model to generalise knowledge from one language to another, thus reducing the need for extensive language-specific labelled data.

Cross-lingual NER models bring with them also limitations. Difficulties arise when models come across domain-specific named entities, terminology or contexts; such issues may prevent a model from generalising effectively.

The lack of labelled data is also an issue and could result in fine-tuning resource constraints. In addition, assuring an aligned representation across languages can be challenging, especially for languages that present different syntactic structures or linguistic features.

Finally, parallel data could not be available for all languages, limiting the models' capabilities since some approaches could rely on parallel text or bilingual dictionaries for alignment.

2.3 Mainly Used Datasets

2.3.1 CoNLL2002 and CoNLL2003

The CoNLL-2002 dataset (Tjong Kim Sang, 2002) covers two languages in the field of named entity recognition, Spanish and Dutch, while the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003) concerns English and German. Each language includes a training file, a development file and a test file. The data contains four types of named entities: persons (PER), organisations (ORG), locations (LOC) and miscellaneous names (MISC).

The Spanish data was taken from articles obtained from Agencia EFE, a Spanish multimedia news agency. The articles are from May 2000. The Dutch data was extracted from four issues (2nd June 2000, 1st July 2000, 1st August 2000 and 1st September 2000) of "De Morgen", a Belgian newspaper.

Tables 2 and 3 show the number of sentences and lines contained in each data file for the Spanish and Dutch data, respectively.

Table 2: Number of sentences and lines in each data file for the Spanish language

Spanish data	Sentences	Lines
Training set	8324	273037
Development set	1916	54837
Test set	1518	53049

Source: Tjong Kim Sang, 2002

Table 3: Number of sentences and lines in each data file for the Dutch language

Dutch data	Sentences	Lines
Training set	15807	218737
Development set	2896	40656
Test set	5196	74189

Source: Tjong Kim Sang, 2002

The English data was taken from the Reuters Corpus, which includes Reuters news stories written between 1996 and 1997. However, the training and the development set consist of ten days' worth of data taken from the files corresponding to August 1996. The texts dating December 1996 were chosen for the test set whilst the preprocessed raw data covers September 1996.

The German data was taken from the ECI Multilingual Text Corpus. This corpus consists of texts in many languages. The data used was extracted from the German newspaper Frankfurter Rundshau. In addition, training, development and test sets were taken from articles dated to August 1992 while the raw data were taken from the months of September to December 1992.

Tables 4 and 5 show the size of the data files for the English and German languages, while Tables 6 and 7 show the number of named entities in each data file for the English and German language respectively.

Table 4: Number of articles, sentences and tokens in each data file for the English language

English data	Articles	Sentences	Tokens
Training set	946	14,987	203,621
Development set	216	3,466	51,362
Test set	231	3,684	46,435

Source: Tjong Kim Sang and De Meulder, 2003

Table 5: Number of articles, sentences and tokens in each data file for the German language

German data	Articles	Sentences	Tokens
Training set	553	12,705	206,931
Development set	201	3,068	51,444
Test set	155	3,160	51,943

Source: Tjong Kim Sang and De Meulder, 2003

Table 6: Number of named entities per data file for the English language

English data	LOC	MISC	ORG	PER
Training set	7140	3438	6321	6600
Development set	1837	922	1341	1842
Test set	1668	702	1661	1617

Source: Tjong Kim Sang and De Meulder, 2003

Table 7: Number of named entities per data file for the German language

German data	LOC	MISC	ORG	PER
Training set	4363	2288	2427	2773
Development set	1181	1010	1241	1401
Test set	1035	670	773	1195

Source: Tjong Kim Sang and De Meulder, 2003

2.3.2 REFLEX

The REFLEX (Research on English and Foreign Language Exploitation) language packs (Simpson et al., 2008) are the outcome of a program sponsored by the American government having the goal of creating basic language resources for less commonly taught languages (LCTLs). The project comprehended the construction of overall language packs for 19 LCTLs in total.

The 19 languages taken into consideration for the project are Amazigh (Berber), Bengali, Hungarian, Kurdish, Pashto, Punjabi, Tamil, Tagalog, Thai, Tigrinya, Urdu, Uzbek, Yoruba, Amharic, Burmese, Chechen, Guarani, Maguindanao (Philippines) and Uighur (China).

Table 8 summarises the number of tokens present in the language packs. The languages analysed have been distinguished between large languages, which contain a larger amount of data (more than 14 million tokens), and small languages, which contain a smaller amount of data (less than 14 million tokens).

Table 8: LCTL's Language Packs

Large Languages	**Small Languages**	**Monolingual Text Tokens**
Urdu	-	14,804,000

Thai	-	39,700,000
-	Bengali	2,640,000
-	Tamil	1,112,000
-	Punjabi	13,739,000
-	Hungarian	1,414,000
-	Yoruba	363,000
-	Tagalog	774,000
-	Tigrinya	617,000
-	Pashto	5,958,000
-	Uzbek	790,000
-	Kurdish	2,463,000
-	Berber	181,000

Source: Simpson et al., 2008

2.3.3 LORELEI

The LORELEI (Low Resource Languages for Emergent Incidents) language packs (Strassel et al., 2016) have been designed to improve the performance of technologies for low-resources, with an emphasis on the use case of resource deployment in unexpected crises such as natural disasters.

The text language packs include alongside data and annotations lexicons and grammatical resources for 23 representative languages (RL) as well as 12 incident languages (IL), listed in Table 9. IL packs are designed to represent the type of information that could be accessible when dealing with an incident involving a language with low resources. The former was chosen to supply typological coverage, while the latter allowed a correct and effective test of the technology's performance. IL packs are designed to represent the type of information that could be accessible when dealing with an incident involving a language with low resources.

2.3.4 WikiANN

The WikiANN dataset (Pan et al., 2017) was built employing the linked entities present in 282 languages available in Wikipedia articles. It was annotated with persons (PER), organisations (ORG), and locations (LOC) tags. The goal of this dataset was the development of a cross-lingual name tagging and linking framework for all the languages available.

2.4 Cross-lingual NER Techniques

The following models have been chosen among the top 30 articles, listed according to relevance on Google Scholar, between 2016 and 2023. They will be divided into three categories: dataset-based techniques, embedding-based techniques and advanced techniques.

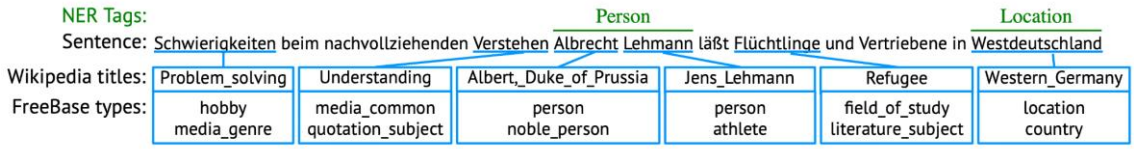
2.4.1 Dataset-based Techniques

Dataset-based techniques depend on labelled or annotated datasets, for training to automatically recognise and categorise named entities in text.

2.4.1.1 Wikification

Cross-Lingual Wikification (Tsai et al., 2016) is a method whose goal is "(..) grounding words and phrases of non-English languages to the English Wikipedia (..)" (Tsai et al., 2016, pg. 222). It is possible to create a connection between words mentioned in texts written in languages different from English and their corresponding entries in the English Wikipedia.

Fig. 9: example of a cross-lingual wikifier on a German sentence



Source: Tsai et al., 2016

For every word in the target language, the authors query Wikipedia in order to identify possible entities or at least align them with the source translation (Fig. 9).

Incorporating cross-lingual Wikification into NER models is especially relevant due to its ability to improve entity recognition in multilingual contexts. This approach completes traditional NER models, like the one proposed by Ratinov and Roth (2009), by expanding the features used. The model encompasses a range of standard features, including those listed by Ratinov and Roth (2009), alongside the use of gazetters as features that draw from titles found in the multilingual Wikipedia, in addition to the use of cross-lingual wikifier features.

This method can help people understand better the context of a text when information is expressed across multiple languages.

The model relies on the following datasets: CoNLL2002/2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for English, German, Spanish, and Dutch, REFLEX (Simpson et al., 2008) for Bengali, Tagalog, Tamil, and Yoruba, and LORELEI (Strassel, 2016) for Turkish.

2.4.1.2 Effective Annotation and Representation Projection (EARP)

This research by Ni et al., 2017 explores three distinct models to solve the problem of limited human-annotated data. These models are Conditional Random Fields (CRFs) and Maximum Entropy Markov Models (MEMMs). The former allows sequential data labelling by modelling dependencies between adjacent labels in a sequence, while the

latter, with a probability structure similar to CRFs, offers an alternative point of view on modelling sequential data.

In addition, the authors propose two distinct neural network architectures: the first (NN1) uses word embeddings as input for capturing semantic relationships between words in a dense space, while the second one (NN2) introduces a smoothing prototype layer. This layer computes the cosine similarity between a word's embedding and predefined prototype vectors learned during training. Subsequently, the weighted average of these vectors is used as input, adding context to the NER process.

The authors want to make sure that the words in the source and target languages have the same embedding space; this allows them to identify the entities in the target languages through translation from the source languages. So, they propose to project the source and target embeddings and use the nearest neighbour approach to align them. The training is performed with a classification loss.

This approach uses annotation projection and representation projection techniques and is weakly supervised.

- Annotation projection generates weakly labelled NER training data in the target language by projecting annotations from a source language. It aligns comparable corpora or translations to connect source and target language entities.

- Representation projection transfers knowledge from a source language to the target language using shared word embeddings. It maps word representations from the target language to the source language, allowing the source-language NER system to be directly applied to the target language without re-training. The method also incorporates co-decoding schemes to increase the model's effectiveness.

Without heavily depending on human annotation in the target languages, this method offers a solid foundation for creating efficient cross-lingual NER systems.

The approach is validated using four target languages: Japanese, Korean, German and Portuguese. Finally, they performed a case study on the Ugyhur language.

2.4.1.3 Zero-Resource Cross-Lingual Named Entity Recognition

This study by Bari et al., 2020 introduces an unsupervised cross-lingual NER model to facilitate training for the target language using labelled data from the source language.

It is made use of two distinct encoders: one dedicated to the source language and the other to the target. Furthermore, the source model uses a bidirectional Long Short-Term Memory Conditional Random Field (LSTM-CRF) architecture established by Lample et al., 2016.

Subsequently, it is applied a two-step process to transfer this base model to the target language. Initially, the authors use word-level adversarial training to project monolingual word embeddings into a shared space and create preliminary cross-lingual links. Then, the joint training of the target model in accordance with the source model is improved by an augmented fine-tuning method.

Adversarial training is generally implemented in generative adversarial networks (GANs), particular architectures composed of two sub-networks: a generator and a discriminator. The latter is responsible for detecting if inputs come from generated distributions or real data, while the former is responsible for generating outputs that are as real as possible to "fool" the discriminator.

Once we have an embedding layer which is shared among languages, we can use the same classification layer to classify the entities as the words in the source and target languages are represented in the same way. An interesting aspect of this approach is the fact that it does not rely on any language alignment of the dataset and translation software.

The model has been carried out on five target languages: Spanish, Dutch, German, Arabic, and Finnish. English is the source language, and its NER-tagged sentences are

drawn from the CoNLL-2003 shared task dataset (Sang and Meulder, 2003). Meanwhile, the CoNLL-2002 shared task dataset (Sang, 2002) provides data for Spanish and Dutch. For Finnish, the NER dataset is obtained from Ruokolainen et al., 2019, with slight tag modifications. Lastly, the AQMAR Arabic Wikipedia Named Entity Corpus (Mohit et al., 2012) is used for Arabic.

2.4.1.4 Cheap Translation for Cross-Lingual Named Entity Recognition

The authors focus on translating annotated data from a high-resource language to a low-resource language using lexicon. This method, called Cheap Translation (Mayhew et al., 2017), is based on the fact that lexicon is considerably more affordable and accessible compared to parallel text resources.

The training data is built thanks to the translation of source data into the target language. This process shares similarities with phrase-based statistical machine translation systems like MOSES (Koehn et al., 2007).

The fundamental model used is Ratinov and Roth, 2009 which incorporates standard features such as forms and capitalisation, among others. The methodology includes the training of Brown clusters using entire Wikipedia dumps for respective languages, a technique applicable to any monolingual corpus. Moreover, the approach embraces the multilingual gazetteers and wikifier features proposed by Tsai et al.,2016.

When compared to state-of-the-art techniques using established benchmark datasets, this approach shows appreciable improvements. Additionally, this method's portability is greatly enhanced by its simplicity and resource efficiency, which distinguish it from all earlier approaches used in the field.

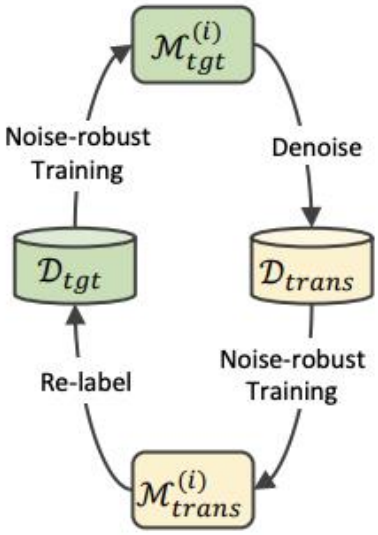
The model relies on the following datasets: CoNLL2002/2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for English, German, Spanish, and Dutch, REFLEX (Simpson et al., 2008) for Bengali, Tamil, and Yoruba, and LORELEI (Strassel, 2016) for Turkish and Hausa.

2.4.1.5 Collaborative Label Denoising Framework (CoLaDa) for Cross-Lingual Named Entity Recognition

Commonly employed methods for cross-lingual NER are often associated with challenges stemming from label inaccuracies due to faulty translation and label projection or limitations within the models themselves.

The approach outlined by Ma et al., 2023 introduces a model-collaboration-based denoising strategy aimed at addressing label inaccuracies caused by faulty translation and label projection. This method entails training models on both data sources and iteratively employing them to cleanse the pseudo-labels from both sources. Initially, a model M_{tgt} , trained on pseudo-labelled target-language data (D_{tgt}), is utilised to refine the translation data derived from label projection. Subsequently, an enhanced model M_{trans} is used to re-label the unlabeled target-language data (D_{tgt}), aiming to mitigate noise within the data and leading to the enhancement of M_{tgt}. This iterative process results in mutual enhancements across data sources and models, establishing a progressive enhancement cycle depicted in Figure 10.

Fig. 10: CoLaDa approach functioning



Source: Ma et al., 2023

Furthermore, it is observed that tokens sharing similarity in the feature space can contribute to denoising. According to the principles of anomaly detection (Gu et al., 2019), the proximity of a data point to its neighbours indicates the presence of anomalous behaviour. If a token's label notably contradicts the labels of neighbouring tokens, it is plausible that the token's label contains noise.

Based on this premise, the authors propose the adoption of an instance-collaboration-based denoising approach. This strategy uses the label consistency within the neighbourhood of each token in the feature space to recalibrate the significance of soft-labelled examples during knowledge distillation. The Collaborative Label Denoising framework (CoLaDa) integrates this instance-collaboration approach into the model-collaboration denoising scheme,

The CoLaDa framework effectively tackles the challenges of label noise in cross-lingual NER. The model-collaboration denoising approach combines models from different data sources to enhance the quality of each other's labelling and amplifies overall learning. The instance-collaboration strategy further reinforces the denoising process by accounting for label consistency among tokens with similarities, resulting in more accurate soft-labelled examples for knowledge distillation.

It is important to note that the applicability of the framework depends on the presence of both a translation system and unlabeled data in the target language. Consequently, its application might be restricted for languages lacking unlabeled text or translation resources. Moreover, the effectiveness of the knowledge distillation step relies on access to a sufficient quantity of unlabeled text.

The model relies on the following datasets: CoNLL2002/2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for English, German, Spanish, and Dutch and WikiAnn (Pan et al., 2017) for Arabic, Hindi, and Chinese.

2.4.2 Embedding-based Techniques

Embedding-based techniques are a class of methods used in Natural Language Processing where words, phrases, or other data elements are transformed into continuous vector representations, known as embeddings.

2.4.2.1 Bilingual Word Embedding Translation (BWET)

Xie et al., 2018 have proposed a novel technique that applies bilingual word embeddings to improve word mapping across languages and the use of self-attention to deal with difficulties in word order variation.

A lexical mapping approach combines the strengths of discrete dictionary-based methods and continuous embedding-based methods. The process begins with the projection of embeddings from various languages into a shared embedding space. Following this, discrete word translations are derived by identifying the nearest neighbours within this projected space. The next step involves training a model using the translated data, and this improves the resource efficiency of embedding-based techniques and the alignment advantages of dictionary-based methods.

Additionally, the authors integrate an order-invariant self-attention mechanism into the neural architecture to reduce differences in word order during unsupervised cross-lingual NER transfer. This mechanism permits the reorganisation of information within encoded sequences, effectively accounting for variations in word order between the source and target languages.

These two approaches result in achieving state-of-the-art or competitive outcomes in cross-lingual NER tasks for commonly evaluated languages. Significantly, this approach necessitates fewer resources compared to previously used methodologies.

The model relies on the following datasets: CoNLL2002/2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for English, German, Spanish, and Dutch.

2.4.2.2 Unifying Model Transfer and Data Transfer (UniTrans)

UniTrans (Wu et al., 2020) unifies model transfer and data transfer techniques while using insights from unlabeled target-language texts to enhance the cross-lingual NER process.

A voting scheme generates pseudo-hard labels for unlabeled target-language data and introduces supervision from both soft labels and the newly introduced pseudo-hard labels. This scheme has been adopted to enhance knowledge distillation within UniTrans.

The model demonstrates exceptional performance. Notably, the potential of UniTrans is further elevated through teacher ensembling, showcasing its versatility and ability to achieve state-of-the-art results.

The model relies on the following datasets: CoNLL-2002 (Tjong Kim Sang, 2002) for Spanish and Dutch, CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) for English and German, as well as NoDaLiDa-2019 (Johansen, 2019) for Norwegian.

2.4.2.3 Dynamic Gazetteer Integration

The method introduced by Fetahu et al., 2022 employs a token-level gating layer to enhance pre-trained multilingual transformers with gazetteers containing named entities from a specific target language or domain. The entity knowledge from gazetteers is selectively used, activated only when the textual representation of a token proves insufficient for the NER task.

The NER approach intends to achieve two main objectives. Firstly, the challenge of multilingualism is addressed by encoding sentences through the pre-trained XLMR model (Conneau et al., 2020). Secondly, to account for variations across domains, XLMR is enriched with multilingual gazetteers. As both components - the XLMR encoding and the gazetteer-based enhancement - offer complementary information, they are combined using the mixture of experts (MoE) methodology (Shazeer et al., 2017).

This empowers the model to dynamically determine the proportion of information required for NER.

Comprehensive evaluations underscore that external gazetteers significantly guide and elevate NER knowledge transfer. Moreover, the quality of training data has a significant impact on the NER model's ability to transfer knowledge effectively in different languages and domains.

In conclusion, the approach presents an effective solution when dealing with the challenge of NER knowledge transfer.

The languages covered include English, Spanish, Dutch, Russian, Turkish, Korean and Farsi.

2.4.3 Advanced Techniques

Advanced Cross-lingual NER techniques use advanced strategies, such as meta-learning and consistency training, to improve performance in scenarios where traditional techniques might struggle due to resource limitations.

2.4.3.1 Meta-Learning for Cross-Lingual Named Entity Recognition

Wu et al., 2020 consider the scenario where one source language has a great amount of annotated data while the target languages do not.

The model is developed from a recently introduced model-agnostic meta-learning approach (Finn, Abbeel, and Levine, 2017). The main goal is to facilitate effective NER adaptation across languages despite having minimal or no labelled data available in the target languages.

The approach involves constructing pseudo-meta-NER tasks using labelled data from the source language. The meta-learning algorithm determines the optimal initialisation of model parameters, enabling swift adaptation to new tasks. During the adaptation

phase, each individual test instance is treated as a distinct task. A task-specific pseudo-training set is created, and the model pre-trained through meta-learning is fine-tuned accordingly.

To enhance the model's generalisation capabilities, a masking scheme is introduced, which reduces the model's reliance on entities and encourages predictions based on contextual cues. Furthermore, the loss function is augmented with an additional term that enforces a maximum constraint. This modification directs the model's attention to specific tokens, minimising the potential for mispredictions. Consequently, the transfer of meta-knowledge associated with these mispredictions to target languages is mitigated.

The proposed approach significantly outperforms existing state-of-the-art methods.

The model relies on the following datasets: CoNLL-2002 (Tjong Kim Sang, 2002) for Spanish and Dutch, CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) for English and German, Europeana Newspapers (Neudecker, 2016) for French and MSRA (Cao et al., 2018) for Chinese.

2.4.3.2 Dual-Contrastive Framework for Low-Resource Cross-Lingual Named Entity Recognition (ConCNER)

Fu et al., 2022 introduce a dual-contrastive framework called ConCNER, which is designed to address the challenges related to dealing with limited labelled data in the source language. ConCNER has two distinct objectives to address different grammatical levels: Translation Contrastive Learning (TCL), which aligns sentence representations within translated sentence pairs, and Label Contrastive Learning (LCL), which focuses on aligning token representations within the same labels.

Moreover, it also applies the knowledge distillation method, which involves using the NER model trained previously as a teacher to guide the training of a student model on unlabeled target-language data, enhancing its alignment with the target language.

Through a series of experiments conducted on widely used datasets, it is demonstrated that the proposed ConCNER framework outperforms existing methods, showcasing competitive performance in cross-lingual NER tasks with limited labelled data in the source language.

However, it is important to acknowledge a potential limitation of the framework; each label is regarded as an individual unit in contrastive learning without considering potential relationships between different labels.

The model relies on the following datasets: CoNLL2002/2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for English, German, Spanish, and Dutch and WikiAnn (Pan et al., 2017) for Arabic, Hindi, and Chinese.

2.4.3.3 Prototype Knowledge Distillation Network (ProKD)

Through the unsupervised prototype knowledge distillation network (ProKD), Ge et al., 2023 address the challenges in zero-resource cross-lingual NER.

ProKD makes use of the prototype concept meant as representative instances of the data distribution. ProKD's approach involves two key techniques: first, it uses a method based on contrastive learning to align prototypes' representations of two languages, and by adjusting the distances between prototypes in both the source and target languages embedding spaces, the model enhances the teacher network's ability to acquire knowledge that's not limited by language barriers.

Then, it employs a prototypical self-training approach. This means the student network is retrained using distance information from prototypes. This helps the student network "understand" the language's structure and acquire language-specific knowledge.

In conclusion, ProKD ensures the teacher network learns broadly applicable knowledge, and the student network becomes more language aware. This unique approach helps NER when there's not much-labelled data in the target language.

The model relies on the following datasets: CoNLL2002/2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for English, German, Spanish, and Dutch and WikiAnn (Pan et al., 2017) for Arabic, Hindi, and Chinese.

2.4.3.4 Consistency Training for Cross-lingual Named Entity Recognition (ConNER)

ConNER is a consistency training framework designed by Zhou et al., 2022 in order to improve the performance of cross-lingual NER by addressing the challenges of limited data availability in target languages, particularly in zero-shot learning.

ConNER comprises two components:

- Translation-based Consistency Training on Unlabeled Target-Language Data involves training the model on unlabeled target-language data, taking advantage of a translation-based consistency approach. The objective is to enhance cross-lingual adaptability by ensuring the model makes consistent predictions.

- Dropout-based Consistency Training on Labeled Source-Language Data passes the same labelled source-language sample through the model twice, prompting the model to generate consistent probability distributions for the same token from two separate dropout operations.

Through the applications of these techniques, it is able to achieve the goal of using unlabeled target-language data while minimising the risk of overfitting the source language. However, the method's feasibility depends on the availability of machine translation models or systems, which may not always be easily accessible.

The model relies on the following datasets: CoNLL2002/2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for English, German, Spanish, and Dutch and WikiAnn (Pan et al., 2017) for Arabic, Hindi, and Chinese.

Consistency Training for cross-lingual NER will be examined more in-depth in the next chapter.

Chapter III

Application of the ConNER Model to the Italian Language

3.1 Task Description

In this chapter, our focus is on evaluating the effectiveness and accuracy of the ConNER model when applied to four different languages: English, Spanish, German, and French. The model is pre-trained in English, French, Spanish and German and tested in a zero-shot NER task on an Italian dataset. Our ultimate goal is to assess its reliability and performance for the Italian language.

All the information about the ConNER model was taken from the original research paper "ConNER: Consistency Training for Cross-lingual Named Entity Recognition" by Zhou et al., 2022.

3.2 ConNER Original Dataset

The authors of ConNER make use of three different datasets: CoNLL02 and CoNLL03 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) and WikiAnn (Pan et al., 2017). The first two cover four languages: Dutch, German, Spanish, and English, while the third includes English, Chinese, Arabic, and Hindi.

In every experiment, the BIOES entity annotation system was used for entity annotation. The BIOES system is based on the earlier BIO (Begin, Inside, Outside) format (Ramshaw and Marcus, 1995), this scheme offers a systematic method for classifying entities as "B" -"Beginning," "I" - "Inside," "O" - "Outside," "E" - "End," or "S" - "Single," according to where considered named entity is positioned in the sentence. We can see an example in Fig. 11.

Fig. 11: Example of BIOES Entity Annotation System

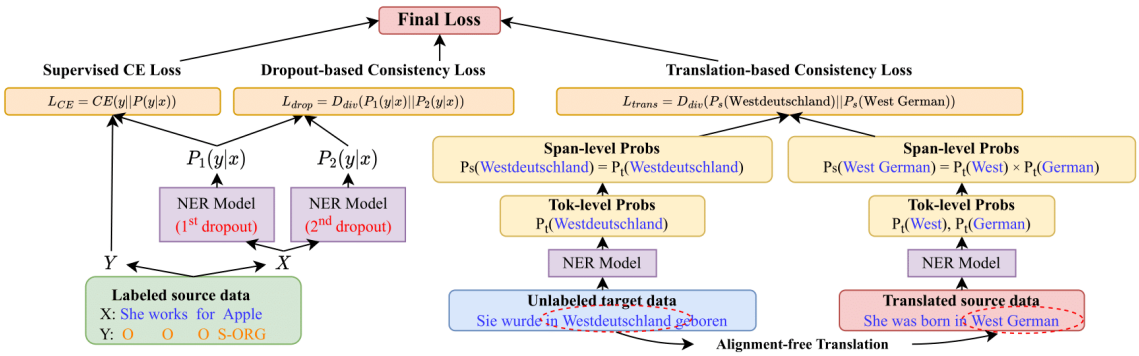
```
Alex S-PER
is O
going O
with O
Marty B-PER
A. I-PER
Rick E-PER
to O
Los B-LOC
Angeles E-LOC
```

3.3 Methods

In the paper by Zhou et al., 2022, the primary aim is to enhance cross-lingual NER by leveraging unlabeled target-language data. Instead of conventional methods, the paper adopts a robust technique known as consistency training (Miyato et al., 2018). This approach seeks to improve the model's reliability and generalisation ability by making its output distributions more consistent.

Previous studies have explored various consistency training approaches in NER, both at the token-level and sequence-level. Token-level methods, such as introducing Gaussian noise (Zheng et al., 2021) or replacing words (Lowell et al., 2020), aim to make models more resilient to noise and variations in the data. However, these methods face challenges when noisy tokens have different labels than the original ones. Some attempts at back-translation-based consistency techniques (Wang and Henao, 2021) encountered difficulties with word alignment across different languages, resulting in issues with maintaining consistency in entity recognition. To address these challenges, alternative methods have incorporated constituent-based tagging schemes (Zhong et al., 2020).

Fig.12: ConNER model framework



Source: Zhou et al., 2022

ConNER aims to ensure that predictions remain consistent, across languages. It tackles word alignment issues without relying on tools and effectively handles variations in numbers during translation. To this end, the authors propose to optimize three main objectives (Fig. 12):

- **Supervised Classification:** The labelled data in the source language is fed to the NER model and a Cross Entropy Loss is employed to train the classification capabilities. In particular, given an input text X and its corresponding label Y, the model predicts $\hat{Y}$ as:

$$\hat{Y} = NERModel(X)$$

  and the classification loss is calculated using cross entropy, as in most classification tasks:

$$LCE = CrossEntropy(Y, \hat{Y})$$

- **Dropout-based Consistency**: During training, each instance is passed twice through the model using two different dropout regularization on the neurons. The two outputs are then compared through the Kullback-Leibner (KL) divergence to enforce model stability across small text variations. Indeed, by minimizing the KL divergence, the authors encourage the model to output

similar distributions when the input is slightly perturbed. More in detail, given an input text X, the two output distributions are obtained as follows:

$$\hat{Y}1 = NERModel(X \mid dropout1)$$
$$\hat{Y}2 = NERModel(X \mid dropout2)$$

the **Dropout Consistency Loss** is then obtained as

$$Ldrop = Div(\hat{Y}1||\hat{Y}2)$$

Where Div is defined as

$$Div(\hat{Y}1||\hat{Y}2) = \frac{1}{2}(KL - Div(\hat{Y}1||\hat{Y}2)) + KL - Div(\hat{Y}1||\hat{Y}2))$$

and

$$KL - Div(P, Q) = Sum(Pi\ log2(Pi/Qi)$$

- **Translation-based Consistency**: This is the core of the proposed approach. Unlabeled data in the target language is translated to the source language. The two versions are then fed to the model, which predicts the entity in the two languages. Finally, a third loss is employed to minimize the KL divergence between the output distributions in the target and source languages. This process encourages the model to similarly represent instances across different languages and attribute them with the same entities. Specifically, given a text input in the target language Xt, the source language translation is obtained

$$Xs = Translate(Xt)$$

Successively, the target and source entity probabilities are obtained as

$$Yt = NERModel(Xt)$$
$$Ys = NERModel(Xs)$$

Yt and Ys are then aligned to match same entities and the KL-Divergence is again used to enforce the similarity between the distributions.

3.4 Motivations for Model Selection

There are several reasons why we decided to choose the ConNER model, here we list the most significant:

1. ConNER has demonstrated its effectiveness in solving translation issues. This is achieved through translation-based consistency training, which simplifies the translation process without relying on word alignment tools. Moreover, it effectively handles variations in numbers throughout translations. This methodology proves efficient when utilizing unlabeled data in the target language.

2. Furthermore ConNER successfully integrates labeled source language data with target language data during training. This integration enhances generalization capabilities, across languages.

3. We were impressed by the ConNER framework because it introduces a novel approach to consistency training. The ConNER model was showcased at the 2022 Conference on Empirical Methods in Natural Language Processing. This shows its relevance.

4. One notable feature is its ability to validate and replicate results. The provided code and resources are easily accessible, comprehensible and openly available.

5. This particular model was chosen for analysis due to its applications, which include enhancing multilingual search engines and automating translation services, for information extraction.

3.5 Experiment

3.5.1 Environment

The project was developed using Google Colaboratory, a cloud-based computing environment. More specifically, a V100 High VRAM GPU was used as part of the computational resources available for this project.

3.5.2 Repositories

In our research on Google Colaboratory, we used external code repositories to carry on our work.

- **Zhou et al.'s ConNER Repository**: This repository contains the code and resources related to the original project by Zhou et al., 2022. We relied on this repository to understand and implement the ConNER model, which was the focus of our research.

- **"pytorch-neural-crf" Repository**: It includes important components and functions that allow us to integrate neural Conditional Random Fields (CRF) into the model using the PyTorch framework.

3.5.3 Software Requirements

We need a set of software and library requirements to execute our project.

- **Python => 3.7:** To use the latest language features and be compatible with libraries, Python 3.7 is required. Python, a versatile and widely used programming language, and underpins our ConNER project's codebase. Python has been chosen for its large package library and machine-learning capabilities.

- **PyTorch => 1.7:** PyTorch 1.7 is required to use its latest features. ConNER requires PyTorch, a popular deep-learning framework. The library is used to develop and train the ConNER model.

- **Transformers => 3.5:** The ConNER project heavily relies on the Hugging Face Transformers library, specifically version 3.5 or above. The library offers pre-trained transformer-based models and tools for developing various NLP tasks. This approach relies on transformer-based models for feature extraction and contextual understanding of text. The Transformer architecture was introduced in the previous chapter.

- **conlleval = 0.2:** We use the 0.2 version of "conlleval", which is used to standardise our ConNER model's performance evaluation. This tool evaluates every outcome using the CoNLL-2002 assessment script. A certain iteration of "conlleval" ensures uniformity throughout the evaluation process, guaranteeing that the evaluation results match named entity recognition benchmarks.

Please note that compliance with these requirements is necessary to duplicate and expand our ConNER-based project tests.

3.5.4 Dataset Used

In our project we decided to use the MultiNERD dataset (Tedeschi and Navigli, 2022) which is developed from the work of two earlier collaborative research initiatives, WikiNEuRal (Tedeschi et al., 2021) and NER4EL (Tedeschi et al., 2021)

MultiNERD is enhanced with cutting-edge techniques for creating low-quality data, which were modelled after WikiNEuRal. Although automatic techniques have achieved a level of annotation accuracy and have covered various languages, they have primarily focused on coarse-grained entities and relied solely on Wikipedia as a textual source. In contrast high-quality data were predominantly centered around the English language and lacked disambiguation information.

Additionally, MultiNERD incorporates components from NER4EL, particularly its proficiency in entity linking and fine-grained entity classes.

The MultiNERD dataset includes ten languages: Chinese, Dutch, English, French, German, Italian, Polish, Portuguese, Russian, and Spanish. This is an impressive range of languages, which makes the dataset flexible and a helpful resource.

It was manually annotated and it expands the types of NER categories such as Person (PER), Location (LOC), and Organization (ORG). It adds named entities, such as Animal (ANIM), Biological Entity (BIO), Celestial Body (CEL), Disease (DIS), Event (EVE), Food (FOOD), Instrument (INST), Media (MEDIA), Plant (PLANT), Time (TIME), and Vehicle (VEHI). Moreover, the dataset includes text from WikiNews and Wikipedia.

To be used in this project, the dataset was converted to the CoNLL format. In accordance with CoNLL norms, entities that are not LOC, ORG, or PER are classified as Miscellaneous (MISC). The entity position labelling is changed, with the labels "B" and "E" being swapped out for "B", "I" - Intermediate and “E” labels.
The model was trained on a portion of the MultiNERD dataset, with 2265 train sentences for English, 2298 for Spanish, 2133 for German and 2693 for French.

Data processing is done using specialised scripts provided by Zhou et al. and found in the project repository. Sentence translation is done using the Facebook Translator on the Hugging Face platform. This translator is called NLLB-200 and it was developed by the NLLB Team in 2022. The NLLB 200 model is mainly used for machine translation research for low-resource languages. It allows the translation of sentences across a range of 200 languages. The introduction of NLLBs could greatly assist in translating languages within communities. Consequently, there could be the possibility of sharing knowledge and diverse cultural aspects with an audience, both within and beyond their community (NLLB Team, 2022). We decided to opt for this model with respect to Google Translate since it is an open-source model and a cost-effective solution and this guarantees accessibility.

3.5.5 Preprocessing and Data Handling

For the experiments, we use the Italian language as target, and we employ all the other languages as source. In particular, the model only uses the NER labels from the source languages. Indeed, the NER labels of the target language are only used to evaluate the performances of the models. Due to time and computational limitations, as previously mentioned, we only make use of about 2000 samples from each language.
The training procedure requires the presence of the translation of the training set through Google Translate. Given the cost of this last tool, we opted for another translation utility from the Hugging Face Repository. In particular, we used the NLLB-200 model, which is a transformer-based translation model which supports over 200 languages (NLLB Team, 2022).

3.5.6 Evaluation Metrics

The evaluation of the results is based on the model accuracy and averaged micro-F1 score related to each named entity. Thanks to these metrics, we are able to gain insights and understand completely the performance of the model.

1. **Accuracy:** By computing the accuracy of the model we are able to assess the overall model performance. It measures the proportion of correctly identified named entities in relation to the total number of named entities in the dataset (1). A high accuracy score indicates that the model categorises and classifies correctly named entities across various languages.

$$(1)\ Accuracy = \frac{TN\ +\ TP}{TN\ +\ TP\ +\ FN\ +\ FP} = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

2. **F1 Score:** F1 scores are particularly significant because they consider both precision and recall (2). In this way, it is possible to understand the model's ability to identify specific named entity types. By calculating F1 scores for each

named entity category, the study can pinpoint areas where the model excels and areas where it may need improvement.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$(2)\ F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

3.5.7 Training Details

In this section, we report the details of the training phase for the application of ConNER to the Italian language.

For each source language (English, French, Spanish and German) and the target language (Italian), we produce the training set (train.txt), translations (trains.txt), unlabelled data (unlabel.txt), development set (dev.txt) and test set (test.txt) files. In the files are obtained in the following way:

- train.txt and dev.txt are directly obtained from the train and test files of MultiNERD for all the source languages;
- trans.txt is obtained using the translation script from the ConNER repository, which we modified in order to use the novel translation tool;
- unlabel.txt is obtained from the preprocess script from the ConNER repository.
- test.txt is obtained from the test set file of MultiNERD for the Italian language.

As indicated by the authors we employ the XLM-RoBERTa-large (Conneau et al., 2020) with CRF head (Lample et al., 2016) as base model for the ConNER.

Finally, we follow the training procedure of Zhou et al., 2022 by training the NER model for 10 epochs and selecting the best checkpoint using the predefined development set. The model is evaluated on the target-language test set. We report the micro-F1 score.

3.6 Results

The ConNER framework is evaluated on the CoNLL dataset using different pairs of languages. For each model, the authors evaluate the Accuracy and the F1 Score.

This method achieves relevant performances compared with its competitors posing itself as one of the state-of-the-art methods for Multi Language NER.

The authors also validate the proposed strategy through an ablation study demonstrating the effectiveness of the three optimisations. In the next sections, we evaluate this method to a novel set of languages which, to the best of the author's knowledge, have not been explored so far in this particular domain.

Table 9: Values of Accuracy and F1 scores for every named entity in percentage and the number of phrases used to train every model.

Model	Accuracy (%)	F1 (%)					Train Phrases
	Overall	Overall	LOC	MISC	ORG	PER	-
En2It	96.36	72.28	89.20	33.86	86.64	93.10	2265
Es2It	96.46	73.55	91.39	35.54	**88.64**	**94.30**	2298
De2It	92.06	50.85	83.47	29.72	32.07	60.60	2133
Fr2It	**97.33**	**80.75**	**92.38**	**61.57**	82.82	93.94	2693

The English to Italian (En2It) model performs remarkably well in terms of overall accuracy, achieving 96.36%. This high level of accuracy is complemented by an F1 score of 72.28%, indicating a balanced trade-off between precision and recall. Notably, the model excels in recognising locations (89.20%) and persons (93.10%), making it particularly suitable for tasks involving these named entity categories.

Moving on to the Spanish to Italian (Es2It) model, we can observe a great overall performance. This model achieves an accuracy of 96.46% and an F1 score of 73.55%,

similarly to the En2It model. It stands out in recognising organisations (88.64%) and persons (94.30%), reaching the highest F1 scores for these entities among all models.

In contrast, the German to Italian (De2It) model exhibits a lower overall accuracy of 92.06% and a relatively modest F1 score of 50.85%. Despite having a reasonable training dataset size of 2133 phrases, this model faces challenges in recognising organisations (32.07%) and persons (60.60%). These results suggest that the transfer of knowledge from German to Italian for NER may require further refinements to improve performance.

Finally, the French to Italian (Fr2It) model emerges as the standout performer in this experiment. It achieves the highest overall accuracy of 97.33% and an impressive F1 score of 80.75%. This model benefits from a substantial training dataset containing 2693 phrases. Its exceptional performance extends to miscellaneous entities (61.57%) and locations (92.38%). These results underline the model's robustness in transferring knowledge from French to Italian, making it a highly effective tool in this context.

3.7 Conclusions

All models perform particularly well in recognising locations as well as persons, showing robustness in these categories. In addition, the performance in recognising organisations remains at a reasonable level. However, it is important to underline that miscellaneous entities tend to have a low F1 score since this category includes all those entities that are not either PER, LOC or ORG.

In order to understand these results better, we have to consider the origins of the source languages with respect to the target language. Spanish, French and Italian are also known as Romance languages since they have all originated from Latin, and their linguistic structures as well as vocabulary reach a high degree of similarity. Since the Fr2It model had the best performance, we could argue that French and Italian are the closest in similarity among all the source languages.

If we consider the En2It model, it scores the third-best results of all four models since the English language originates from Germanic languages, but it was still influenced by French and Latin during its development.

Lastly, the De2It model shows the lowest scores. This is no surprise since German is not a language that originated from Latin.

In conclusion, our research gives insights into transferring NER capabilities from English, French, Spanish and German languages to Italian. We emphasise the importance of selecting the right source language considering its origin and addressing challenges related to each named entity category.

We hope to have brought a contribution to the field of cross-lingual NER and provided practical guidance for researchers and professionals aiming to use these techniques in diverse linguistic settings.

.

Bibliography

Abney, S. “Understanding the Yarowsky Algorithm.” Computational Linguistics, no. 3, MIT Press - Journals, Sept. 2004, pp. 365–95.

Beel, J., et al. TF-IDuF: A Novel Term-Weighting Scheme for User Modeling Based on Users’ Personal Document Collections. 2016.

Churchill, G. “Stochastic Models for Heterogeneous DNA Sequences.” Bulletin of Mathematical Biology, no. 1, Springer Science and Business Media LLC, 1989, pp. 79–94

Devlin, J., et al. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. 2018.

Eddy, S. R. “Profile Hidden Markov Models.” Bioinformatics, no. 9, Oxford University Press (OUP), 1998, pp. 755–63

Fort, S., et al. Exploring the Limits of Out-of-Distribution Detection. 2021.

Haffari, G., and R. Sarkar. Analysis of Semi-Supervised Learning with the Yarowsky Algorithm. AUAI Press, 2007.

Kejriwal, M. Domain-Specific Knowledge Graph Construction. Springer, 2019.

Kim Sang, Erik F. Tjong. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. 2002.

Kobayashi, M., and K. Takeda. “Information Retrieval on the Web.” ACM Computing Surveys, no. 2, Association for Computing Machinery (ACM), 2000, pp. 144–73.

Leidner, J. L., et al. “Grounding Spatial Named Entities for Information Extraction and Question Answering.” Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References, 2003.

Majeed, A., and I. Rauf. “Graph Theory: A Comprehensive Survey about Graph Theory Applications in Computer Science and Social Networks.” Inventions, no. 1, MDPI AG, 2020, p. 10

Medhat, W., et al. “Sentiment Analysis Algorithms and Applications: A Survey.” Ain Shams Engineering Journal, no. 4, Elsevier BV, Dec. 2014, pp. 1093–113.

Mikolov, T., et al. Efficient Estimation of Word Representations in Vector Space. 2013.

Mirza, P. Extracting Temporal and Causal Relations between Events. 2016.

Mollà, D., et al. “Named Entity Recognition for Question Answering.” Proceedings of the Australasian Language Technology Workshop, 2006, pp. 51–58.

Nabil Alkholy, E. M., et al. “Question Answering Systems: Analysis and Survey.” International Journal of Computer Science & Engineering Survey, no. 06, Academy and Industry Research Collaboration Center (AIRCC), 2018, pp. 1–13.

Olivetti, E. A., et al. “Data-Driven Materials Research Enabled by Natural Language Processing and Information Extraction.” Applied Physics Reviews, no. 4, AIP Publishing, 2020.

Palatucci, M., et al. Zero-Shot Learning with Semantic Output Codes. 2009.

Rivera-Trigueros, I. “Machine Translation Systems and Quality Assessment: A Systematic Review.” Language Resources and Evaluation, no. 2, Springer Science and Business Media LLC, 2021, pp. 593–619.

Robertson, Stephen. “Understanding Inverse Document Frequency: On Theoretical Arguments for IDF.” Journal of Documentation, no. 5, Emerald, 2004, pp. 503–20.

Spärck Jones, Karen. “IDF Term Weighting and IR Research Lessons.” Journal of Documentation, no. 5, Emerald, 2004, pp. 521–23.

Tedeschi, S., et al. “Named Entity Recognition for Entity Linking: What Works and What’s Next.” Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, 2021.

Vaswani, A., et al. Attention Is All You Need. 2017.

Voorhees, E. M. “Natural Language Processing and Information Retrieval.” Information Extraction, Springer Berlin Heidelberg, 1999, pp. 32–48.

Wang, Y., et al. “Nested Named Entity Recognition: A Survey.” ACM Transactions on Knowledge Discovery from Data, no. 6, Association for Computing Machinery (ACM), July 2022, pp. 1–29.