Université Sorbonne Paris Nord

École doctorale Galilée
Laboratoire d'Informatique de Paris Nord

Information Extraction For Arabic Language and its Dialects

Présentée par Niama EL KHBIR

Thèse de doctorat d' INFORMATIQUE

Soutenue publiquement le 22/11/2024 devant le jury composé de :

Thierry CHARNOIS, Prof. HDR	Université Sorbonne Paris Nord	Directeur
Nadi TOMEH, MCF	Université Sorbonne Paris Nord	Encadrant
Alexis NASR, Prof. HDR	Université Aix Marseille	Rapporteur
Farah BENAMRA, Prof.	IRIT-Toulouse University	Rapporteuse
Houda BOUAMOR, Assoc. Prof.	Carnegie Mellon University in Qatar	Examinatrice
Hanene AZZAG, Prof. HDR	Université Sorbonne Paris Nord	Examinatrice

Abstract

The exponential growth of textual data has underscored the importance of Information Extraction (IE) as a crucial subfield of Natural Language Processing (NLP). This PhD thesis advances the state of the art in IE, with a particular focus on the Arabic language and its dialects, which present unique challenges due to their rich morphological structure, diacritic variability, and dialectal diversity. The research is organized around three major contributions.

Firstly, we introduce the first neural joint information extraction system designed specifically for Modern Standard Arabic (MSA). This model integrates BERT-based token encoding with Conditional Random Fields (CRFs) for entity and event trigger identification, and Feed Forward Neural Networks (FFNs) for relation and event argument classification. Our model effectively navigates the complexities of Arabic morphology, setting a new benchmark for IE performance in Arabic.

Secondly, the thesis explores Cross-Dialectal Named Entity Recognition (NER) in Arabic. We construct comprehensive datasets for Egyptian, Moroccan, and Syrian Arabic dialects, and explore the transferability of NER models trained on MSA to these dialects in a zero-shot setting. This approach significantly mitigates the challenge of limited annotated resources, enabling broader application of NER across diverse Arabic-speaking regions.

Finally, we propose a novel framework for joint IE tasks, employing Differentiable Beam Search on Graph Recurrent Neural Networks (GRNNs). This method tackles the issue of exposure bias in sequence-to-sequence models, enhancing the model's ability to capture and leverage interdependencies between entities, relations, and events. The approach demonstrates robust performance across multiple languages, including Arabic, providing a versatile tool for multilingual IE.

This thesis not only advances Arabic NLP by addressing its specific linguistic challenges but also contributes to the broader field of multilingual information extraction, offering new methodologies and insights for future research.

Keywords: Joint Information Extraction, Named Entity Recognition, Arabic Language Processing.

Contents

List of Figures

List of Tables

List of Abbreviations

ACE Automatic Content Extraction. 11, 32

CA Classical Arabic. 3

CNN Convolutional Neural Network. 22, 23, 30

CRF Conditional Random Field. 5, 8, 9, 19, 20, 22, 28, 41–44, 81–84, 101, 105

DNN Deep Neural Network. 21

EAE Event Argument Extraction. 1, 2, 100

ED Event Detection. 40, 55

EE Entity Extraction. 1, 2, 100

ETE Event Trigger Extraction. 1, 2, 100

FFN Feed Forward Neural Network. 5, 8, 9, 42, 44, 101

GNN Graph Neural Network. 6, 26

IE Information Extraction. 1–9, 17, 19, 23, 24, 40–42, 54, 58, 59, 79, 100, 102–106

LDC Linguistic Data Consortium. 11, 47

LLM Large Language Model. 5, 6, 23, 24, 104

LSTM Long Short-Term Memory. 27, 28

MEM Maximum Entropy Model. 18, 19

MSA Modern Standard Arabic. iii, 3, 8, 40, 61, 62, 64, 66, 67, 69, 72–77, 79, 100, 103, 104

NER Named Entity Recognition. 8, 15, 17, 19–24, 32, 33, 40, 45, 51, 58, 59, 61–63, 66, 67, 69, 70, 72, 74–80, 97, 100, 103, 104

NLP Natural Language Processing. 1, 3, 5, 6, 14, 17, 23, 30, 59, 61, 76, 79, 102, 106

PLM Pretrained Language Model. 8, 21, 24, 26, 69, 74, 77–79, 91–93, 103

RE Relation Extraction. 1, 2, 40, 55, 58, 100

RNN Recurrent Neural Network. 16, 21, 22, 27, 28, 81, 82, 84, 91, 97, 105

SVM Support Vector Machine. 18, 19, 32

Chapter 1

Introduction

1.1 Information Extraction

With the exponential growth of available textual documents, Information Extraction (IE) has emerged as a crucial subfield of Natural Language Processing (NLP) for efficiently extracting relevant and structured information from unstructured textual data such as scientific papers, newswires, weblogs, medical documents, and so on. There are different subtasks under the broad field of information extraction that focus on extracting different types of information such as named entity recognition, relation extraction, coreference resolution, entity linking, event trigger extraction, and event argument extraction. In this thesis, we focus on four tasks of information extraction namely:

- **Entity Extraction (EE)** is the task of identifying and classifying entities (such as persons, organizations, locations, dates, etc.) mentioned in unstructured text into predefined categories or types.

  In IE, we distinguish between an “entity” which is an object or set of objects in the world, and an “entity mention” which is a piece of text referring to an entity. Mentions can be categorized as named mentions (proper names), nominal mentions (common nouns or noun phrases), and pronoun mentions (pronouns). We also differentiate between “entity extraction”, which involves identifying these mentions and assigning types to them, and “entity linking” which connects these mentions to their corresponding entries in a knowledge base, ensuring accurate linkage to real-world entities.

- **Relation Extraction (RE)** is the task of identifying and classifying semantic relationships or associations between entities.

- **Event Trigger Extraction (ETE)** and **Event Argument Extraction (EAE)** are closely related tasks in IE, integral to event extraction. **ETE** focuses on identifying words within text that indicate the occurrence of events or actions, categorizing them into predefined event types. **EAE** complements ETE by identifying and extracting the arguments associated with these events. Arguments represent entities involved in events, fulfilling specific roles such as “agents” or other defined roles.

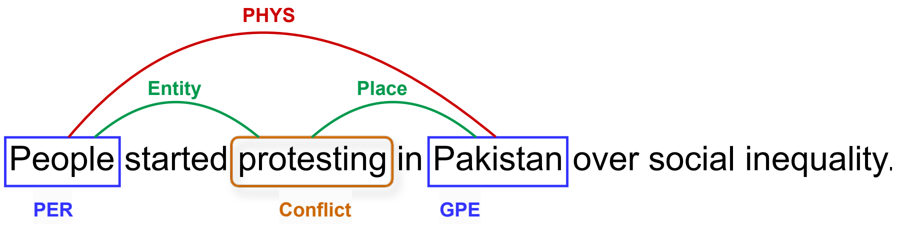

PHYS
Entity
Place
People started protesting in Pakistan over social inequality.
PER
Conflict
GPE

Figure 1.1: Example of an Information Extraction Graph.

Our focus on the four specific tasks of EE, RE, ETE, and EAE is driven by their fundamental importance and interdependence in creating a comprehensive and accurate representation of information in text. These four tasks exhibit notable similarities: The process of ETE mirrors that of EE, both involving the identification and classification of elements in text. Likewise, EAE parallels RE by linking event triggers to one or more associated entities, similar to how RE connects pairs of entities based on semantic relationships.

The process involves transforming raw textual information into a format suitable for automatic analysis. The extracted information can then be used for various purposes, with a wide range of applications, such as tracking news, monitoring public opinion, and identifying emerging trends. In the business world, IE can be used to analyze competitors, extract product details, and find customers. In healthcare, it helps extract medical data, assess disease risks, and personalize treatments. Law enforcement leverages it to extract information from reports, identify suspects, and track criminal activities.

As an illustrative example, consider the following sentence, also depicted in Figure 1.1: "*People started protesting in Pakistan over social inequality*". The task of information extraction consists of extracting the following information:

1. "*People*" is an entity of type `Person` (`PER`);
2. "*Pakistan*" is an entity of type `Geo-political` (`GPE`);
3. "*Protesting*" is an event trigger of type `Conflict`;
4. There is a relation of type `Physical` between "*People*" and "*Pakistan*";
5. "*Pakistan*" is an event argument of type `Place` to the event "*Protesting*";
6. "*People*" is an event argument of type `Entity` to the event "*Protesting*"

In this example, the information can be used to track social unrest, assess risks, and create early warning systems.

In this thesis, we adopt a **structured prediction approach** where a sentence serves as input, aiming to produce a graph structure as output. In this graph structure, entities and event triggers are represented by nodes, while relations and arguments are represented by arcs. This graphical representation enables capturing complex interactions and dependencies between the various entities and events present in the text. In Figure 1.1, entities (nodes) are highlighted in blue, triggers (nodes) in orange, relations (edges) are depicted by red arcs, and arguments (edges) are represented by green arcs.

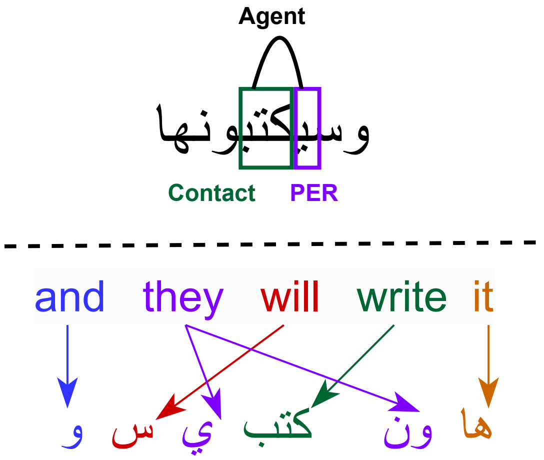


Figure 1.2: Example of Arabic Morphological Complexity.

1.2 Arabic Information Extraction

Motivated by the multiple applications of information extraction, there is a compelling reason to focus specifically on the Arabic language. Arabic is widely spoken by over 420 million people worldwide, with rich linguistic nuances and great cultural significance. The Arabic language is a collection of multiple variants, with Modern Standard Arabic (MSA) being the formal written standard used in media, culture, and education. MSA is based on Classical Arabic (CA), the language of the Qur'an, but is more modern in vocabulary. In contrast, Arabic dialects are the true native language forms, used for informal daily communication. They are not standardized and are primarily spoken, although they are becoming more common in writing due to the rise of electronic communication. Arabic dialects are loosely related to CA and are the result of the interaction between different ancient dialects and other languages that existed in the Arab world (Habash, 2010).

Challenges of Arabic Information Extraction The field of Arabic natural language processing has evolved significantly over the past years, with researchers developing a diverse range of tools and models for various NLP tasks (cf. Section 2.2.5). This includes advancements in text classification, sentiment analysis, and language modeling. However, despite this progress, the area of information extraction research specifically dedicated to Arabic text remains relatively underdeveloped compared to other languages like English or Chinese. Few works have addressed entity extraction (Shaalan and Raza, 2009; Abdallah et al., 2012; Traboulsi, 2009) and relation extraction (Ben Hamadou et al., 2010; Al Zamil and Al-Radaideh, 2014) as independent tasks, with limited efforts directed towards event extraction. However, no research to date has tackled these tasks in an integrated joint manner. This gap stems from the inherent complexities of the Arabic language, which pose significant challenges for developing robust IE systems:

Diacritic Form	عَقْدِنَا	عُقَدِنَا	عِقْدِنَا	عَقَّدْنَا
Buckwalter Transliteration	EaqodinaA	EuqadinaA	EiqodinaA	Eaq adonaA
Translation	Our contract	Our pyschoses or Our knots	Our necklace or Our decade	We complicated

Table 1.1: Example of Arabic Diacritic Variation.

- **Morphological Complexity**: Arabic features a rich morphological system with complex derivational morphology and non-concatenative processes. This means a single root word can have numerous derivations with varying lengths and grammatical functions. For example, the verbal sentence "وسيكتبونها" (*wsyktbwnhA* [1]*; and they will write it*) showcases the complexity of morphology for IE tasks in several ways. As shown in Figure 1.2, this sentence can be annotated for 3 information:

  1. The letter "ي" as an entity of type `Person`;
  2. The subword "كتب" as an event trigger of type `Contact`;
  3. The subword "ي" as an event argument of type `Agent`.

  Traditional tokenization methods that simply split text at whitespace struggle with these variations, making it difficult to identify the core meaning and grammatical role of each word in the context of IE tasks.

- **Diacritic Variation**: Arabic script uses diacritics, which are small marks above or below letters, to distinguish sounds and grammatical features. These diacritics are crucial for accurate understanding, as the absence of a single diacritic can completely change the meaning of a word. Table 1.1 presents an example of diacritic variation for the word "عقدنا" (*EqdnA*). The word is displayed with different diacritic forms, each representing a different interpretation.

  Missing diacritics in Arabic can drastically alter meaning posing a significant challenge for IE models that rely solely on the surface form of the text.

- **Dialectal Diversity**: Arabic exists in a wide range of dialects, each with distinct vocabulary, grammar, and pronunciation. This diversity presents a challenge for developing generalizable IE models that work across different dialects. For example, the word for "*bread*" in modern standard Arabic is "خبز" (*xbz*). In Egyptian Arabic, the word for bread is "عَيْش" (*Eayo$*), while in Tunisian Arabic, it is" طابونة" (*TAbwnp*). This diversity presents a challenge for developing generalizable information extraction models that work across different dialects.

- **Limited Annotated Resources**: Compared to languages like English, Arabic suffers from a scarcity of high-quality, annotated datasets for IE tasks. The annotation process, which involves manually labeling text data with the desired information, is expensive

[1] We use the Buckwalter (Buckwalter) transliteration scheme for Romanization.

and time-consuming. This limited data availability hinders the training and evaluation of robust NLP models for Arabic IE. Although some resources exist (Mohit et al., 2012; Walker and Consortium, 2005; Moussa and Mourhir, 2023), they primarily focus on entity extraction tasks or are limited to Modern standard Arabic.

Our Approach Given the complexities and unique challenges posed by the Arabic language in the context of information extraction, it is essential to develop tailored solutions that can handle these issues and advance the field. Addressing these challenges is critical for several reasons. Firstly, the rich morphological structure and extensive use of diacritics in Arabic require sophisticated models capable of parsing and understanding nuanced linguistic features. Without investing in such models, it will be challenging to achieve accurate and reliable IE from Arabic text. Secondly, the diversity of Arabic dialects presents a significant challenge. Developing adaptable systems that can generalize across different spoken forms of Arabic is necessary to create comprehensive language technologies that serve the entire Arabic-speaking world. Thirdly, the scarcity of annotated datasets calls for innovative approaches to data generation and model training. Investing in the creation of annotated resources and the development of models that can perform well despite limited data is essential for advancing Arabic IE.

Methods used for IE in English can often be adapted for Arabic, although some modifications are necessary due to the unique characteristics of the Arabic language. Many foundational tools used in processing English text have Arabic versions or equivalents. For instance, word embeddings like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) have been trained on Arabic corpora, and advanced models like BERT (Devlin et al., 2019) have an Arabic version known as AraBERT (Antoun et al., 2020). These tools provide robust representations of Arabic text, facilitating the application of machine learning models to Arabic NLP tasks. Neural network architectures used for English can be applied to Arabic as well. These models are language-agnostic at their core, meaning they can process any input text as long as it is correctly preprocessed. For instance, tokenization, stemming, and handling of diacritics are preprocessing steps that need to be adapted for Arabic to ensure that the models can accurately interpret the text.

While Large Language Models (LLMs) like GPT-4 (Radford and Narasimhan, 2018) have shown remarkable capabilities in various NLP tasks, our approach for Arabic IE uses BERT for word representation and simpler architectures such as Conditional Random Fields (CRFs) and Feed Forward Neural Networks (FFNs). This choice is due to several reasons. Firstly, LLMs require extensive computational resources, making them impractical for many research environments and applications with limited hardware. Additionally, prompting LLMs for IE tasks is challenging and does not always guarantee good performance. BERT captures rich contextual information without the high costs and complexities of LLMs. Secondly, combining BERT with CRFs and FFNs enables efficient and targeted modeling. CRFs are excellent for sequence labeling tasks such as named entity recognition, while FFNs offer simplicity and speed for less complex tasks. This balance between performance and efficiency suits the specific challenges of Arabic information extraction. Lastly, our focus on these models leverages BERT's robust contextual embeddings while maintaining manageable complexity, ensuring the development of practical, high-performing IE systems for Arabic, even with resource-intensive alternatives like LLMs available. Furthermore, we leverage transfer learning for named entity recognition in Arabic dialects, in resource-constrained settings.

By addressing these issues, we make significant contributions to the field of Arabic information extraction and facilitate the development of more accurate and comprehensive language technologies for Arabic. Detailed insights about our contributions are presented in Section 1.4.

1.3 Interdependent Information Extraction Tasks

In the context of this thesis, we adopt a **structured prediction paradigm** where a sentence serves as input, and our systems generate a graph structure as output. In this graph, entities and event triggers are represented by nodes, while relations and arguments are represented by arcs. This graphical representation effectively captures the complex interactions and dependencies between the various elements present in the text.

Alongside the challenges related to the Arabic language, an important challenge of general information extraction is developing models that can capture and leverage the interdependence across different IE tasks.

Limitations of Pipeline Approaches Traditional approaches to IE involved pipeline structures (Chan and Roth, 2011), treating tasks in a sequential manner where each model (for entity, relation, and event extraction) is trained separately. The entity extraction task would be performed first, with its output (extracted entities) fed into the relation extraction task, and so on. These pipeline approaches suffered from several limitations. First, errors made in earlier stages propagate through the pipeline, leading to compounded inaccuracies and ultimately hindering overall performance. Second, valuable contextual information extracted in one task is not readily available for the others, resulting in suboptimal performance due to the lack of integrated task interdependence.

Rise of Joint Information Extraction Due to these limitations, recent advancements have focused on joint information extraction, leveraging multitask learning. These models tackle all information extraction tasks simultaneously, by using for example shared word vector representations, or by employing parameter sharing, which consists in sharing the weights or biases of one or more layers of the neural network. One prominent category of joint models is graph-based architectures, which leverages the output graph structure for dependency modeling, considering entities and event triggers as graph nodes and relation and event arguments as graph edges. Techniques include Graph Neural Networks (GNNs) and dynamic message passing approaches (Luan et al., 2019; Wadden et al., 2019).

However, predicting the entire graph in joint information extraction is both computationally expensive and challenging. This complexity stems from the absence of predefined constraints that guide other NLP tasks, such as well-defined grammatical rules. For instance, syntactic parsing benefits from linguistic syntax, which offers clear guidelines and reduces ambiguity in sentence structure. In contrast, information extraction must infer relationships and dependencies from diverse and often ambiguous text, demanding sophisticated models to accurately capture these nuances.

Alternatively, sequential and auto-regressive frameworks have emerged as robust alternatives in joint IE. These approaches (Miwa and Bansal, 2016; Yu et al., 2019) treat information extraction as a sequence labeling problem, where each token in the input sequence is labeled based on its role (e.g., entity type, relation, or event trigger). Popular architectures for this approach include Recurrent Neural Networks (Rumelhart et al., 1986) and their variants or Transformers (Vaswani et al., 2017).

However, an unexplored area in IE research is the direct modeling of the IE graph using auto-regressive frameworks. Unlike traditional approaches that process the whole input sentence sequentially, we propose a novel methodology. We linearize the IE graph and use auto-regressive methods for labeling, enhancing accuracy through a beam search procedure during inference. By treating the entire IE graph as a cohesive unit, our approach aims to capture intricate interdependencies and enhance the coherence of extracted information.

Challenges of Sequential Prediction in Joint Models While these models have shown success in joint IE tasks, their reliance on sequential prediction introduces limitations. During training, these models are typically optimized by maximizing the locally normalized likelihood of each token in the reference (gold standard) sequence given the labels of previous reference tokens. Essentially, the model learns by comparing its predictions to the "correct" answers and adjusts its internal parameters accordingly.

A crucial discrepancy emerges during inference. The model no longer has access to the previous correct labels; it relies solely on its own predictions from earlier steps in the sequence. This mismatch between training with complete information and inference with potentially erroneous predictions is known as exposure bias, which can significantly impact model performance.

Researchers have proposed various techniques to address this issue, including: (1) Schedule Sampling (Bengio et al., 2015): This technique gradually introduces the model's own predictions during training, mimicking the inference scenario and reducing exposure bias. (2) Curriculum Learning (Bengio et al., 2009): This approach involves progressively increasing the difficulty of training tasks, starting with simpler problems and gradually introducing more complex ones.

Our Approach Additionally, exposure bias can manifest in models that employ different training and inference strategies. For instance, our proposed model described in the previous paragraph uses beam search during inference, a strategy not explicitly seen during training. To address this limitation, recent research has introduced training objectives that incorporate the search process. These methods use continuous approximations of beam search Goyal et al. (2018), making the search procedure differentiable. This compatibility with gradient-based learning allows the model to understand its decoding behavior during training, leading to improved performance in tasks like named entity recognition and segmentation. However, this approach hasn't been widely explored for more complex IE tasks like graph generation, where relations between extracted entities become crucial.

In this context, we propose an IE framework, where the four tasks are framed as a sequence labeling problem, using auto-regressive models for labeling the linearized IE graph, and beam search as a decoding procedure, along with a differentiable beam search version during training.

1.4 Contributions and Publications

In this thesis, we briefly present our contributions that address the challenges discussed in Sections 1.3 and 1.2.

1.4.1 Contributions to Arabic Information Extraction

We contribute to Arabic information extraction through two significant works:

- **ArabIE: Joint Entity, Relation, and Event Extraction for Arabic**

  To address the limitations of existing Arabic information extraction systems, which have primarily focused on named entity recognition, and also address the morphological complexity of Arabic, we propose the first neural joint information extraction system for the modern standard Arabic exploring different modelizations for the Arabic text. In this model, we use BERT as a token encoder, then we perform IE in two steps: a node (entity and trigger) identification step using CRFs, and an edge (relation and argument) classification step using FFNs. Our results present a baseline for future research on Arabic information extraction and show comparable performance to state-of-the-art models for other languages such as English, Spanish, and Chinese.

  Publication: Niama El Khbir, Nadi Tomeh, and Thierry Charnois. 2022. ArabIE: Joint Entity, Relation and Event Extraction for Arabic. *In Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 331–345, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

- **Cross-Dialectal Named Entity Recognition in Arabic**

  To address both the dialectal diversity and the scarcity of manually annotated resources for Arabic dialects, we study the transferability of named entity recognition models between Arabic dialects. We construct four datasets, including a MSA dataset and datasets for Egyptian, Moroccan, and Syrian Arabic. By training a span-based Named Entity Recognition (NER) model on top of a Pretrained Language Model (PLM) encoder on the MSA data, we demonstrate that the model can achieve acceptable performance on the other datasets in zero-shot settings, requiring no additional annotation effort. This work has the potential to enable the use of NER for a wider range of applications in the Arabic context.

Publication: Niama Elkhbir, Urchade Zaratiana, Nadi Tomeh, and Thierry Charnois. 2023. Cross-dialectal Named Entity Recognition in Arabic. *In Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP)*, Singapore (Hybrid). Association for Computational Linguistics.

With these two contributions, we aim to enhance language technologies dedicated to Arabic, and facilitate advancements in areas such as information extraction from Arabic text data.

1.4.2 Contributions to IE tasks Interdependence

We contribute to general information extraction through the following work:

Information Extraction with Differentiable Beam Search on Graph RNNs

The model we developed for Arabic IE is based on two main components: A CRF for the step of node (entity and trigger) identification and a FFN for the step of edge (relation and argument) classification. This model makes decisions on nodes and edges independently, not taking into account the dependencies between the different elements. To adequately model these dependencies, we propose a novel approach that casts graph generation as autoregressive sequence labeling using beam search. Such an approach presents differences in training and decoding frameworks, which makes it prone to the exposure bias problem, which we address by making this same model training aware of the decoding procedure using a differentiable version of beam search. Our approach outperforms non-decoding-aware methods on a variety of datasets across different languages, demonstrating its effectiveness in addressing exposure bias and improving IE accuracy.

Publication: Niama El Khbir, Nadi Tomeh, and Thierry Charnois. 2024. Information Extraction with Differentiable Beam Search on Graph RNNs. I*n Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9084–9096, Torino, Italia. ELRA and ICCL.

1.5 Thesis Outline

Chapter 2 lays the groundwork for the research presented in this thesis. It introduces core concepts and task definitions. Following this foundation, the chapter delves into a comprehensive review of recent advancements in the field of information extraction. This review encompasses relevant areas like multitask learning and decoding strategies, establishing a strong theoretical foundation for the proposed research.

Shifting the focus to Arabic text, Chapters 3 and 4 delve into the specific challenges of information extraction in this language, including dialectal variations. These chapters present the proposed methods for tackling these challenges and achieving effective information extraction in Arabic.

Chapter 5 broadens the scope to general information extraction tasks beyond Arabic text. It details the proposed method for information extraction applicable to a broader range of languages, addressing the challenge of interdependent information extraction tasks.

Chapter 2

Background And Literature Review

2.1 Task Definitions

This section focuses on four core tasks within information extraction that are particularly relevant to our work: entity extraction, relation extraction, event trigger extraction, and event argument extraction. Figure 2.1 provides a visual overview of information extraction tasks, highlighting the tasks we address in blue. For these specific task definitions, we follow the definitions of the Automatic Content Extraction (ACE)[1] program developed by the Linguistic Data Consortium (LDC) [2].

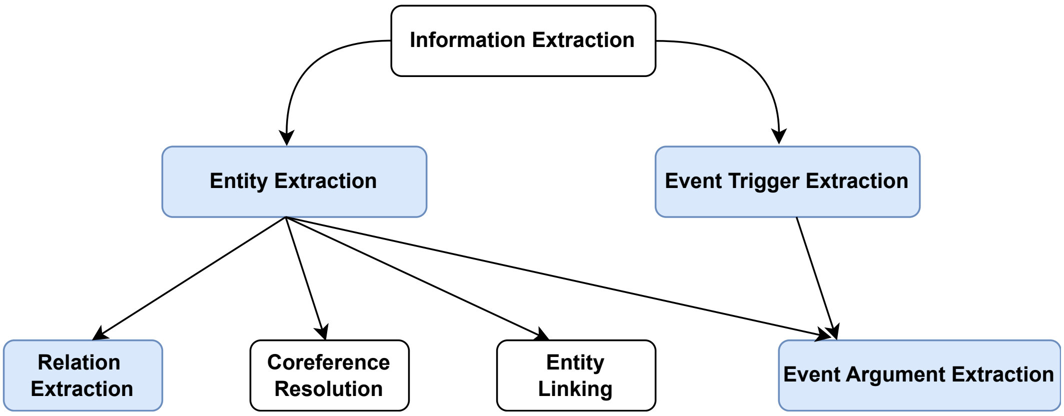


Figure 2.1: Information Extraction Tasks.

We leverage the information extraction task definitions established by the ACE program for several reasons. Firstly, ACE definitions provide a standardized set of terms and task specifications, ensuring clarity and facilitating comparisons with existing research. Secondly, adopting these widely recognized definitions demonstrates familiarity with established practices and contributes to a common ground within the information extraction community.

[1] `https://www.ldc.upenn.edu/collaborations/past-projects/ace`

[2] `https://www.ldc.upenn.edu/`

2.1.1 Entity Extraction

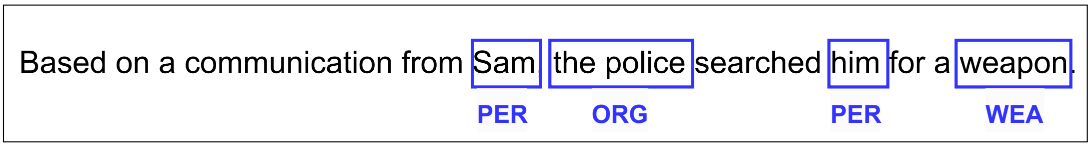
Entity: An entity is an object or a set of objects in the world that can be distinctly identified and classified. Entities can range from individuals and organizations to locations, products, and more.

Entity mention: An entity mention is a segment of text that refers to an entity. These mentions are critical for understanding the specific instances of entities within a text.

Mention type: Entity mentions can be categorized based on how they refer to the entity. Common types include:

- Named Mention (NAM): This refers to the entity by its proper name, such as "*Albert Einstein*", directly identifying a unique entity.
- Nominal Mention (NOM): This uses a common noun or noun phrase to describe the entity, such as "*the scientist*", providing a more general identification.
- Pronoun Mention (PRO): This refers to the entity using a pronoun, such as "*he*" or "*him*", typically relying on the context for identification.

Entity extraction involves the identification and classification of these mentions within a text, assigning each mention its appropriate type. In the example sentence below, there are four entities identified: (1) "*Sam*" identified as a `Person` (NAM), (2) "*The police*" identified as an `Organization` (NOM), (3) "*Him*" identified as a `Person` (PRO), and (4) "*Weapon*" identified as a `Weapon` (NOM).


Based on a communication from Sam the police searched him for a weapon.
PER
ORG
PER
WEA

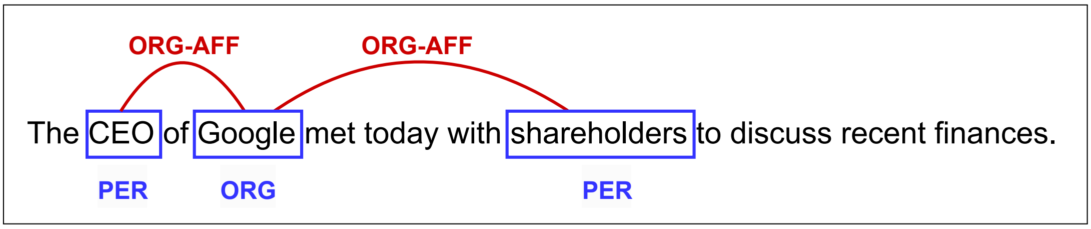
2.1.2 Relation Extraction

Relation: A relation represents a meaningful connection between two entities. The order of the entities and the specific type of relation provide context about the nature of that connection.

Relation type: Relation types categorize broader classes of these connections. Examples include physical relations (`PHYS`), which denote spatial or locational connections, and organizational affiliations (`ORG-AFF`), which indicate professional or hierarchical associations.

Relation subtype: Relation subtypes further refine the specific meaning within each type, offering a more granular understanding of the connection. In the example below, both relations are of type `ORG-AFF`. However, the relation subtype between "*The CEO*" and "*Goole*" is Employment, indicating their employer-employee relationship, while the relation subtype between "*Google*" and "*Shareholders*" is Investor-Shareholder, highlighting the shares ownership aspect.

Additionally, specific constraints often exist on the types of entities that particular relation types can connect. These constraints are essential for maintaining the logical coherence of the extracted relationships, ensuring that the connections made are both meaningful and contextually appropriate. For example, a `Personal-Social` (`PER-SOC`) relation is restricted to connecting entities recognized as persons, such as friends, spouses, or colleagues. This prevents illogical connections, such as linking a person with a location under a social relationship. Table 2.3 provides an example of these constraints, illustrating the permissible entity types for various relation categories.


ORG-AFF
ORG-AFF
The CEO of Google met today with shareholders to discuss recent finances.
PER
ORG
PER

Relation Extraction involves identifying pairs of entities within a text and then assigning them a relation type.

2.1.3 Event Trigger Extraction

Event: An event represents a significant happening or occurrence within a text, often involving participants and a change of state. Events represent specific happenings, situations, or developments.

Event type: An event type is a broad category used to classify events based on their general nature. Examples include `Life` events, related to births, deaths, marriages, etc, and `Movement` events involving movement or transportation.

Event subtype: A more specific categorization within an event type, providing a fine-grained understanding of the event's nature. Subtypes further differentiate events within the same type based on specific characteristics or details. Examples of subtypes within the Life event type might include Birth, Death, and Marriage.

Event trigger: An event trigger is a word or group of words within the event extent (the sentence containing the event) that most explicitly signals the occurrence of the event.

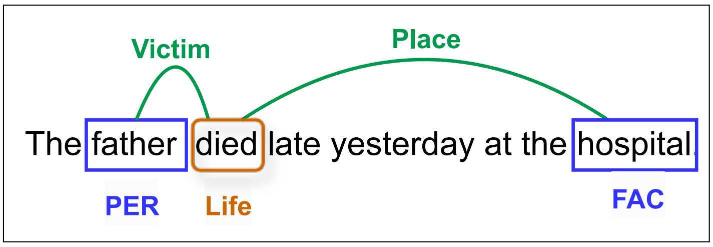
Event trigger Extraction is the process of identifying these event triggers within text data and assigning them appropriate event types. This task is similar to entity extraction in that both involve identifying and categorizing specific elements within a text. Just as entity extraction focuses on recognizing and classifying entities such as people, organizations, and locations, event trigger extraction focuses on identifying and classifying occurrences or happenings.

2.1.4 Event Argument Extraction

Event Argument: An event argument refers to an entity directly involved in the event. The type of involvement, known as the argument role, defines the specific function or relationship the entity has concerning the event trigger. The terms “argument” and “role” are often used interchangeably in this context.

Event Argument Extraction is the process of identifying and classifying all relevant event arguments from text data. This task involves recognizing the entities involved in the event and assigning them appropriate roles based on their relationship to the event trigger. In the example below, “*father*” and “*hospital*” serve as the roles `Victim` and `Place`, respectively, for the event trigger “*died*”, which is categorized under the `Life` event type.

Framework of Events: An event is organized as a structured framework consisting of a trigger and its associated arguments. The event trigger, which signals the occurrence of the event, is linked to various arguments that provide contextual details. These arguments are classified into specific roles that describe their participation in the event. This structured representation is crucial for understanding the dynamics and implications of the event within the text.



Within the framework of this thesis, our focus is exclusively on the extraction of relations and events at the sentence level. It is noteworthy that certain alternative approaches extend this extraction process to the document level for entities (Huang et al., 2021), relations (Xu et al., 2022), and events (Xu et al., 2021).

2.2 Approaches to Information Extraction

In this section, we provide an overview of the different methods and concepts related to information extraction. In Subsection 2.2.1, we explore the diverse learning paradigms used in information extraction dictating how systems acquire the ability to extract specific information from textual sources. In Subsection 2.2.2, we present a chronological review of information extraction techniques, focusing on both individual tasks and potentially joint approaches. We then delve deeper into how these techniques can be classified into main paradigms used for joint information extraction. In Subsection 2.2.5, we present an overview of Arabic NLP and information extraction. Finally, in Subsection 2.2.4, we present some decoding methods usually used for information extraction systems.

2.2.1 Learning Paradigms in Information Extraction

The chosen learning paradigm in information extraction dictates how a system acquires the ability to identify and extract specific information from text data. These paradigms define the overall learning strategy employed by the system. The primary learning paradigms used in information extraction include:

- **Supervised Learning** (Cunningham et al., 2008): This is the dominant paradigm in information extraction due to its effectiveness and the abundance of large, annotated datasets (cf. Section 2.3 for a comprehensive overview of these datasets). In supervised learning, the system is trained on a dataset where each text instance is labeled with the desired output. By analyzing labeled data, the learning algorithm identifies patterns and relationships between text features and corresponding labels. This enables the system to map new, unseen text data to the desired output categories.

- **Unsupervised Learning** (Ghahramani, 2004): Despite not being the primary focus for information extraction tasks due to their reliance on specific information retrieval, unsupervised learning offers valuable tools. It can be employed as a preprocessing step, where techniques like clustering group similar text instances together. This can help identify potential information categories or features relevant to the extraction task. Furthermore, unsupervised approaches are being explored directly for tasks like relation extraction. Research in this area investigates leveraging unaligned parallel text for joint named entity and relation extraction (Munro and Manning, 2012), using Variational Autoencoders (VAEs) to learn latent representations of relations (Yuan and Eldardiry, 2021), and even achieving promising results by inferring relation types solely from named entity types (Tran et al., 2020).

- **Reinforcement Learning**: This paradigm trains a system through a trial-and-error process. The system interacts with the environment (text data in this case) and receives rewards for desired behaviors like correctly extracting information or identifying relevant entities. Over time, the system learns to optimize its actions to maximize these rewards. While reinforcement learning has potential applications in information extraction, particularly for complex tasks or where defining clear labels can be challenging, it is still under development in this field compared to supervised learning. Recent work explores dynamically optimizing extraction order (Huang et al., 2023) and speeding up training with parallel agents (Sharma et al., 2017).

- **Transfer Learning**: Transfer learning leverages knowledge gained from solving one problem and applies it to a different but related problem. In the context of information extraction, pretrained models can be fine-tuned on specific extraction tasks, using the knowledge learned from a vast amount of general text data. This approach can significantly reduce the need for large task-specific datasets and expedite the training process, making it particularly useful when labeled data is scarce or when dealing with domain-specific tasks. Numerous works (Bari et al., 2020; Wu et al., 2020b) transfer NER knowledge from English as a source language to target languages such as Arabic, German, Dutch, Spanish, French and Chinese, with promising results.

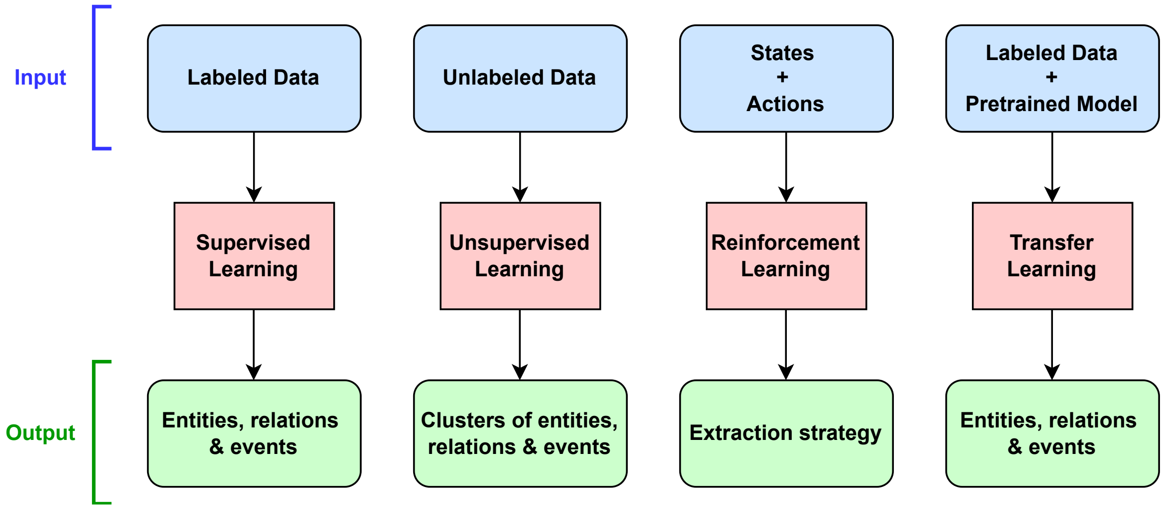


Figure 2.2: Learning Paradigms for Information Extraction.

Other learning paradigms include semi-supervised and weekly-supervised learning. We summarize the main discussed learning paradigms in Figure 2.2. Our research primarily relies on supervised learning (Chapters 3 and 5), the dominant approach due to its effectiveness and the growing availability of large, well-annotated datasets. Supervised learning is well-suited for tasks where achieving high accuracy is crucial. However, for the Arabic dialects where labeled data is scarce (Chapter 4), we explore the potential of transfer learning. Particularly, we leverage the knowledge gained from a vast amount of general Arabic text and adapt it to specific dialectal variations, overcoming the limitations of data scarcity for individual dialects.

2.2.2 A Brief History of Information Extraction

Information extraction has undergone a remarkable transformation over the past few decades. This section explores this evolution by categorizing it into four distinct eras: rule-based, machine learning, deep learning, and the emerging large language model era. While system architectures and trends have evolved, fundamental principles often remain consistent across various tasks. For instance, many information extraction tasks can be framed as classification problems. In binary relation extraction, systems take two entities as input. These entities can originate from the output of a separate entity extraction module, which is common in pipeline systems, or directly from labeled data when focusing solely on the relation extraction task. The task then becomes a classification problem, aiming to identify the specific relation between the entities based on the surrounding text.

Similarly, event trigger extraction involves identifying and labeling sequences of words that correspond to event triggers within the text. Architectures effective for entity extraction, such as Recurrent Neural Networks (RNNs) and transformers, often translate well to event trigger extraction due to their ability to capture and label textual sequences accurately. Following event trigger identification, event argument extraction identifies and classifies entities involved in the event, assigning them specific roles based on their relationship to the event trigger. This process shares similarities with relation extraction, where the focus is on classifying the connections between entities and the identified event trigger.

In the following paragraphs, we will delve deeper into each era, highlighting key methodologies and their impact on the development of IE systems.

Rule-based Systems In the early days of IE, extracting information and automating decisions relied on rule-based systems. These systems do not require complex algorithms or vast amounts of data. Instead, they leverage the domain expertise of human specialists, codified into a set of clear-cut rules. By mimicking human reasoning processes, rule-based and dictionary-based systems excel in well-defined domains, offering explainable results and consistent performance.

This approach excels in well-defined domains like named entity recognition. For instance, a rule-based NER system might process a news article sentence like: "*Barack Obama, the former president of the United States, delivered a speech in Berlin, Germany*". The system uses predefined rules to identify entities based on linguistic patterns. For example, it might recognize a person entity if a sequence starts with a capital letter followed by other capitalized words, or if it matches entries in a list of names. Locations might be identified following prepositions (e.g., "*in*", "*on*") or by matching a list of places, while organizations are identified as capitalized sequences not fitting the person or location patterns. The system iteratively applies these rules to identify entities, ultimately outputting: "*Barack Obama*": `Person`, and "*Berlin, Germany*": `Location`. Early examples include extracting personal names from newspapers (Borkowski and Watson, 1967), company names (Rau, 1991), and LaSIE (Humphreys et al., 1998), which used shallow pattern recognition and lexical patterns.

These methods relied heavily on domain experts to manually define rules for identifying entities, relations, and events. They also depended on manually crafted resources like gazetteers, which are lists of predefined entries, dictionaries, and grammatical rules. While these systems offered explainability and reliability due to their transparent rules and ease of maintenance, they had significant limitations. However, they also have limitations that motivated the focus of the NLP community on machine learning and deep learning approaches. Key limitations included:

- **High Manual Effort**: Creating and maintaining a comprehensive set of rules is time-consuming, labor-intensive, and requires significant domain expertise.
- **Limited Scalability**: Adding new knowledge or handling unforeseen situations might require adding or modifying numerous rules, making the system less scalable for rapidly evolving domains or complex situations.

Shift towards Machine Learning While rule-based systems offered a powerful approach to information extraction in their early days, their limitations led to the exploration of machine learning techniques. These algorithms revolutionized IE by enabling models to automatically learn patterns from data, which allows them to:

- **Adapt to Varying Linguistic Patterns**: Machine learning models can ingest large amounts of text data and uncover the subtle nuances of language. This allows them to handle the complexities of natural language, such as ambiguity, slang, and idiomatic expressions, that often trip up rule-based systems.
- **Enhance Scalability for Evolving Domains**: As new information emerges and language evolves, machine learning models can continuously learn and adapt. This makes them well-suited for domains with constantly changing terminology or vast amounts of data that would be cumbersome to manage with rule-based systems.

Several machine learning architectures have been instrumental in advancements in information extraction. Here are a few prominent examples:

- **Support Vector Machines (SVMs)** (Hearst et al., 1998) is a type of supervised learning models used for classification tasks. The basic idea behind SVMs is to find the optimal hyperplane that separates the data points of different classes with the maximum margin. For entity extraction, there are two main approaches: the binary classification one, which requires training multiple SVMs, one for each entity type, and the multi-class SVM, where words are directly classified into various predefined entity types. For the binary classification approach, let's denote x_i as the feature vector representation of word i in the input text, and y_i as the corresponding label indicating whether word i is part of an entity or not. The goal is to learn a function $f(x)$ that maps each feature vector x_i to its corresponding label y_i. The decision function of an SVM can be represented as:

  $$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b\right)$$

  Where α_i are the Lagrange multipliers, y_i are the class labels (+1 for entity, -1 for non-entity), $K(x, x_i)$ is the kernel function measuring the similarity between feature vectors x and x_i, and b is the bias term.

  During training, SVMs aim to maximize the margin between the hyperplane and the nearest data points (support vectors) of different classes, while minimizing the classification error.

  Related work includes SVMs for entity extraction Takeuchi and Collier (2002); Li et al. (2005), SVMs for relation extraction using lexical, semantic, and syntactic features (Zhou et al., 2005) and SVMs for relation extraction using tree kernels (Culotta and Sorensen, 2004).

- **Maximum Entropy Models (MEMs)** (Bender et al., 2003) are a type of probabilistic graphical models used for classification tasks. MEMs are commonly used for entity extraction due to their ability to handle complex feature representations and capture dependencies between input features and output labels. In the context of entity extraction, let x_i represent the feature vector representation of word i in the input text, and y_i denote the corresponding label. The goal is to learn a conditional probability distribution $p(y|x)$ over the label sequences given the input feature vectors. MEMs achieve this by maximizing the entropy of the model subject to a set of constraints. Mathematically, the probability of a label sequence $\mathbf{y}$ given the input sequence $\mathbf{x}$ can be expressed as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j f_j(\mathbf{y}, \mathbf{x}, i)\right)$$

where $Z(\mathbf{x})$ is the normalization factor ensuring that the probabilities sum to 1, λ_j are the parameters of the model, and $f_j(\mathbf{y}, \mathbf{x}, i)$ are feature functions that capture the compatibility between labels and input features at position i. During training, maximum entropy models aim to learn the parameters λ_j that maximize the log-likelihood of the training data. Once trained, they can be used to predict the most likely label sequence for new input sequences by performing inference. MEMs have been successfully applied to various information extraction tasks, including named entity NER (Chieu and Ng, 2002; Tsai et al., 2005)

- **CRFs** (Lafferty et al., 2001) are probabilistic graphical models used for sequence labeling tasks, making them particularly well-suited for IE tasks like entity extraction, where the order of words is crucial for accurate entity identification. Unlike SVMs that classify words independently, CRFs consider the dependencies between neighboring words. They model these dependencies through a graphical structure, allowing them to leverage contextual information when making predictions.

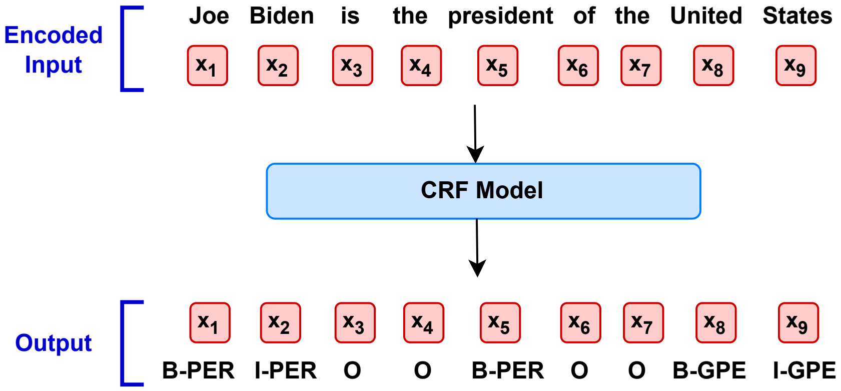

Encoded Input
Joe Biden is the president of the United States
x1 x2 x3 x4 x5 x6 x7 x8 x9
CRF Model
Output
x1 x2 x3 x4 x5 x6 x7 x8 x9
B-PER I-PER O O B-PER O O B-GPE I-GPE

Figure 2.3: Conditional Random Fields for Entity Extraction.

For entity extraction using CRFs with the BIO tagging scheme, let's denote x_i as the feature vector representation of word i in the input text, and y_i as the corresponding label (e.g., "B-PER", "I-LOC", or "O"), following the BIO convention. The labels indicate whether a word is the beginning of an entity (B), inside an entity (I), or outside of any entity (O). The goal is to learn a conditional probability distribution $p(y|x)$ over the label sequences given the input feature vectors. The probability of a label sequence $\mathbf{y}$ given the input sequence $\mathbf{x}$ can be expressed using the CRF model as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j t_j(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i)\right)$$

Where $\mathbf{y}$ is a label sequence, $\mathbf{x}$ is an input sequence, $Z(\mathbf{x})$ is the normalization factor ensuring that the probabilities sum to 1, λ_j are the parameters of the model, and $t_j(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i)$ are feature functions that capture the compatibility between labels $\mathbf{y}_{i-1}$ and $\mathbf{y}_i$ given the input sequence $\mathbf{x}$ at position i.

During training, CRFs aim to learn the parameters λ_j that maximize the log-likelihood of the training data. Once trained, CRFs can be used to predict the most likely label sequence for new input sequences by performing inference using algorithms like the Viterbi algorithm (Forney, 1973). Figure 2.3 illustrates the workflow of a CRF model for entity extraction.

In Chapters 3 and 5, we opt for CRFs for entity and event trigger identification due to their ability to effectively model the sequential nature of labels. Additionally, CRFs excel at handling overlapping entities and complex label dependencies. References such as McCallum and Li (2003); Patil et al. (2020) showcase the successful application of CRFs for NER.

Although machine learning models automatically learn patterns from data and offer greater adaptability, they can inherit biases present in the training data. If the training data disproportionately reflects certain demographics, viewpoints, or writing styles, the model might struggle to generalize well to unseen data and potentially produce biased outputs. Careful data selection, cleaning, and techniques to mitigate bias are crucial for fair and robust information extraction models.

Rise of Deep Learning Recently, deep learning has become the dominant approach due to its ability to automatically learn robust features directly from data. This eliminates the need for extensive manual feature engineering, a time-consuming and knowledge-intensive process that can limit performance. Additionally, deep learning architectures excel at handling complex patterns within text data, leading to potentially superior results. The key aspects of deep learning include enriching input representations and exploring different architectures that we detail below.

Deep learning models for information extraction rely on informative representations of the input text data. These representations often combine various elements to capture word meaning and context:

- **Pretrained Word Embeddings** convert words into vector representations in a high-dimensional space. These include techniques like TF-IDF (Ramos, 2003), which captures the statistical importance of words in a corpus, and word embedding methods like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) which learn vector representations capturing semantic meaning and context based on surrounding words. By providing a pretrained understanding of word meaning and relationships, these embeddings significantly boost model performance.

- **Character Embeddings**, on the other hand, represent words as sequences of vectors, one for each character, enabling the handling of spelling variations and out-of-vocabulary words. There are several ways to create character embeddings, such as one-hot encoding, Character2Vec, and BiLSTM-based methods. Character embeddings offer a more granular way to represent words, allowing models to handle variations and unseen words, ultimately improving performance in tasks like named entity recognition. Related work include character-level taggers for language-independent NER (Kuru et al., 2016), Boosting NER with neural character embeddings (dos Santos and Guimarães, 2015).

- **Contextual Embeddings** leverage PLMs like BERT (Devlin et al., 2019) or RoBERTa to learn how a word's meaning can shift depending on its surrounding words within a sentence. These models are trained on massive amounts of text data. During this training, they learn to analyze the relationships between words and how these relationships influence a word's meaning. By analyzing word relationships during training, these models generate dynamic word representations that capture both core meanings and subtle nuances influenced by context. This enhances information extraction tasks by disambiguating words, improving relationship extraction between entities, and handling complex sentence structures.

- **Additional Features and External Knowledge Sources** can be used to enrich the input representations and guide the learning process, including:
  - **Grammatical Features** such as Part-of-speech (POS) tags, which classify words as nouns, verbs, adjectives, etc., provide valuable clues about the role a word plays within a sentence. This additional information can be particularly helpful for tasks like named entity recognition, where identifying specific word types is crucial.
  - **External Knowledge Sources** such as gazetteer lookup can be integrated into the model. This injects domain-specific knowledge and improves the recognition of specific entities mentioned in the text.

Empowered by these informative input representations, deep learning architectures can be employed to exploit the complex patterns within text data for information extraction tasks. Some of the most widely used approaches include:

- **Deep Neural Networks (DNNs)**: Early attempts used basic DNN architectures for NER tasks Gallo et al. (2008); Lample et al. (2016); Peters et al. (2017). These networks learn feature representations and perform classification in a layered fashion.

- **RNNs**: Due to their inherent ability to handle sequential data like text, RNNs have become a dominant force in NER. They process text one word at a time, allowing them to capture the order and dependencies between words in a sentence. This is crucial for tasks like NER, where recognizing entities often relies on understanding the context of surrounding words. A powerful variant of RNNs, BiRNNs process text in both forward and backward directions simultaneously. This allows them to capture not only the context of preceding words but also the influence of words that follow. This bi-directional understanding significantly improves the accuracy of entity recognition compared to traditional RNNs.

Let x represent a sequence of word representations in a sentence, where x_i is the feature vector representation of word i in the input text and y_i is its the corresponding label. Entity extraction can be formulated as a sequence labeling task, where the goal is to predict the entity label for each word in the input sequence. Given an input sequence x, a basic RNN architecture for entity extraction can be represented as follows:

$$\mathbf{h}_t = \text{RNN}(x_t, \mathbf{h}_{t-1})$$
$$\hat{y}_t = \text{softmax}(\mathbf{W}\mathbf{h}_t + \mathbf{b})$$

where $\mathbf{h}_t$ represents the hidden state of the RNN at time step t, x_t represents the input word embedding at time step t, $\hat{y}_t$ represents the predicted probability distribution over entity labels at time step t, $\mathbf{W}$ and $\mathbf{b}$ are the weight matrix and bias vector of the output layer, and softmax is the softmax activation function.

Related work include BiLSTMs for NER in twitter messages (Limsopatham and Collier, 2016), joint entity and relation extraction from biomedical text (Li et al., 2017).

- **Convolutional Neural Networks (CNNs)**: While less common than RNNs for information extraction tasks, CNNs excel at capturing local patterns within text data. They work by applying filters that scan the text for specific patterns, often short sequences of characters. A typical CNN architecture for entity extraction employs convolutional layers with filters. These filters scan the input sequence looking for specific n-gram patterns. The filters then activate based on the presence or strength of these patterns, capturing local features potentially indicative of entities. Following the convolutional layer, a pooling layer summarizes the most significant features extracted by the filters. Finally, a fully connected layer combines these summarized features and predicts the most likely entity label for each word in the sentence. CNNs have been extensively investigated across various information extraction tasks, demonstrating their versatility and effectiveness. This includes research on joint entity and relation extraction (Adel and Schütze, 2017), relation extraction specifically (Nguyen and Grishman, 2015), and event extraction methodologies (Chen et al., 2015).

- **CRFs**: As discussed earlier, CRFs are probabilistic graphical models that excel at modeling label dependencies within sequences. In the context of NER, this translates to modeling the relationships between named entities within a sentence. By integrating CRFs with deep learning models, we can leverage the strengths of both approaches. Deep learning models excel at feature extraction, while CRFs excel at modeling label dependencies. This combined approach can further improve the accuracy of named entity recognition tasks.

Hybrid Systems While deep learning has become a dominant force in information extraction, there's still room for collaboration between deep learning models and traditional rule-based approaches. Hybrid systems (Zhou and Su, 2002; Florian et al., 2003) aim to leverage the strengths of both techniques. Deep learning models excel at capturing complex patterns and relationships within text data. However, they can sometimes lack interpretability and struggle with limited training data. Rule-based systems, on the other hand, can be highly interpretable and efficient for specific tasks, but they may struggle to adapt to unseen data or complex scenarios. By combining these approaches, hybrid systems can achieve superior performance. Hybrid systems can also combine multiple architectures to perform one task. Examples include models for NER that combine RNNs with CRFs (Lee, 2017), NER models that combine CNNs with CRFs (Strubell et al., 2017), NER models that combine LSTMs with CNNs (Chiu and Nichols, 2016), etc.

Large Language Models for Information Extraction Recently, Large Language Models have become a game-changer in the NLP world and have emerged as a powerful paradigm for IE. These models, often based on transformer architectures, are characterized by their massive number of parameters, often reaching tens or hundreds of billions. This vast training enables them to learn complex patterns from massive amounts of text data, surpassing traditional methods in IE tasks. Two primary paradigms dominate LLM-based IE approaches:

- **Generative Paradigm**: Generative methods exploit the LLM's ability to synthesize target information directly. We can provide the LLM with a carefully crafted prompt specifying the desired output format alongside the raw text itself. For instance, given a scientific paper, an LLM can generate a structured report highlighting experiments, materials, methods, and results, thereby eliminating the need for complex pipelines. Effective **prompt engineering** is crucial in this context, as it guides the LLM towards the desired outcome by specifying target information types, anticipated output formats, and relevant domain-specific knowledge.

  Examples of this approach include GPT-NER (Wang et al., 2023), which transforms NER into a text generation task by adding special tokens to mark entity boundaries, and ChatGPT's application for NER in various contexts (Laskar et al., 2023), demonstrating the model's versatility but also highlighting the need for task-specific adjustments.

- **Discriminative Paradigm**: This approach leverages the power of labeled datasets where text snippets are tagged with the information they contain. By training the LLM on these examples, we equip it to recognize these patterns and classify new unseen text. For example, in extracting key financial indicators from corporate reports, an LLM trained on labeled datasets of revenue, profit, and assets can accurately process new reports and identify these elements. **Domain-specific fine-tuning** further enhances this process by helping the LLM adapt to the unique terminology and information patterns relevant to the target domain. PromptNER (Ashok and Lipton, 2023) uses entity definitions and prompts to identify entities with justifications, showcasing few-shot learning capabilities, while research on relation extraction with GPT-3 and Flan-T5 (Wadhwa et al., 2023) explores various supervision levels and evaluation methods.

Moreover, combining generative and discriminative capabilities can enhance performance. For example, integrating GPT-2's generative abilities with BERT for NER in dialogue systems (Kim et al., 2023) leverages the strengths of both models during training. Data augmentation techniques, such as back-translation and paraphrasing, are also employed to address the challenge of limited labeled data in specific domains, further improving the performance of LLMs.

Despite these advancements, we do not use LLMs for our IE models due to their high computational cost, complexity in fine-tuning, and challenges in interpretability, which complicate error analysis and model improvement.

2.2.3 Main Paradigms for Joint Information Extraction

In the previous section, we explored various architectures for information extraction tasks. While these architectures can be used for individual tasks or combined for multiple tasks at once, a common approach has been the pipeline approach, where the output from one IE subtask is fed into the next subtask. We depict in Figure 2.4 an example of a pipeline system for entity and relation extraction. While this sequential approach has been widely used, it has two major limitations:

- **Error Propagation**: Errors made in earlier stages, like the NER stage, could not be corrected later, when in the RE stage for example, leading to cascading errors and hindering overall performance.
- **Limited Information Sharing**: Valuable contextual information extracted in one task is not readily available for others, hindering the ability to leverage dependencies.

Recognizing these limitations, recent research has focused on joint information extraction frameworks that leverage **multitask learning**. These frameworks tackle all IE tasks simultaneously, allowing for information sharing and potentially achieving superior performance. Common approaches for joint IE include:

1. **Shared Representations**:

   Instead of learning separate word embeddings for each information extraction task, joint models often leverage a common word embedding layer. This layer takes words as input and transforms them into dense vector representations, x_i, that capture their meaning and context within the text. These vector representations are then used by all tasks within the joint model for their specific predictions. Joint models typically employ pretrained word embeddings, such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014), or even more powerful contextualized embeddings like BERT (Devlin et al., 2019) or XLNet (Yang et al., 2020). These pretrained embeddings are derived from large text corpora and encode rich semantic information. They can be fine-tuned during joint model training to capture the nuances relevant to specific IE tasks. Here's how the PLM integration can be mathematically represented:

   $$x_i = f_{LM}(w_i)$$

   where x_i is the dense vector representation for word i in the sentence, w_i is the word itself, and f_{LM} represents the PLM that encodes the word w_i into a vector representation.

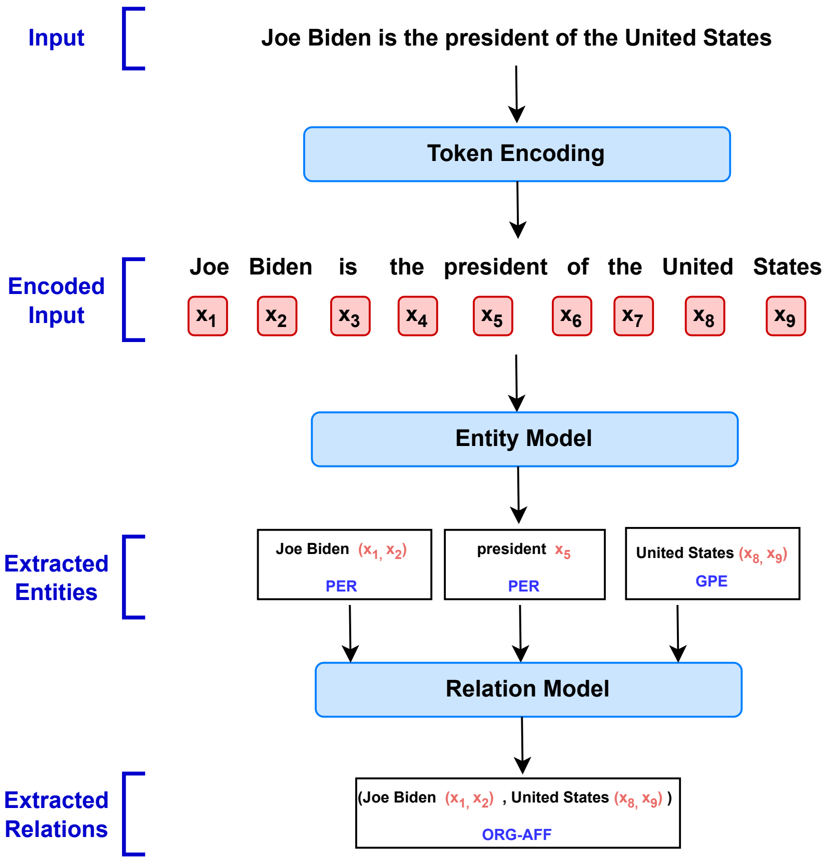

Input
Joe Biden is the president of the United States
Token Encoding
Encoded Input
Joe Biden is the president of the United States
x1 x2 x3 x4 x5 x6 x7 x8 x9
Entity Model
Extracted Entities
Joe Biden (x1, x2) PER
president x5 PER
United States (x8, x9) GPE
Relation Model
Extracted Relations
(Joe Biden (x1, x2) , United States (x8, x9))
ORG-AFF

Figure 2.4: Pipeline System for Entity and Relation Extraction.

2. **Parameter Sharing**:

   In joint models for information extraction, parameter sharing promotes information exchange and reduces model complexity by using the same weights or biases for specific layers across different tasks. This allows the model to learn parameters that benefit multiple tasks simultaneously. There are two main approaches, that we illustrate in Figure 2.5:

   - **Hard Parameter Sharing** (Caruana, 1993): This method uses a single set of weights across all tasks. The model learns a single unified representation for all tasks, typically by sharing hidden layers while maintaining separate task-specific output layers. An example is the works of Miwa and Bansal (2016); Zheng et al. (2017), who use a single model with a unified set of parameters to jointly extract entities and relations.
   - **Soft Parameter Sharing** (Duong et al., 2015): Each task has its own set of weights, but the model encourages them to be similar by penalizing large differences between weights for related tasks. This approach allows for more task-specific learning compared to hard sharing. Relevant works for information extraction include those of Lin et al. (2020); Zhang and Ji (2021); Nguyen et al. (2021a).

   Several studies have explored parameter sharing in joint entity and relation extraction

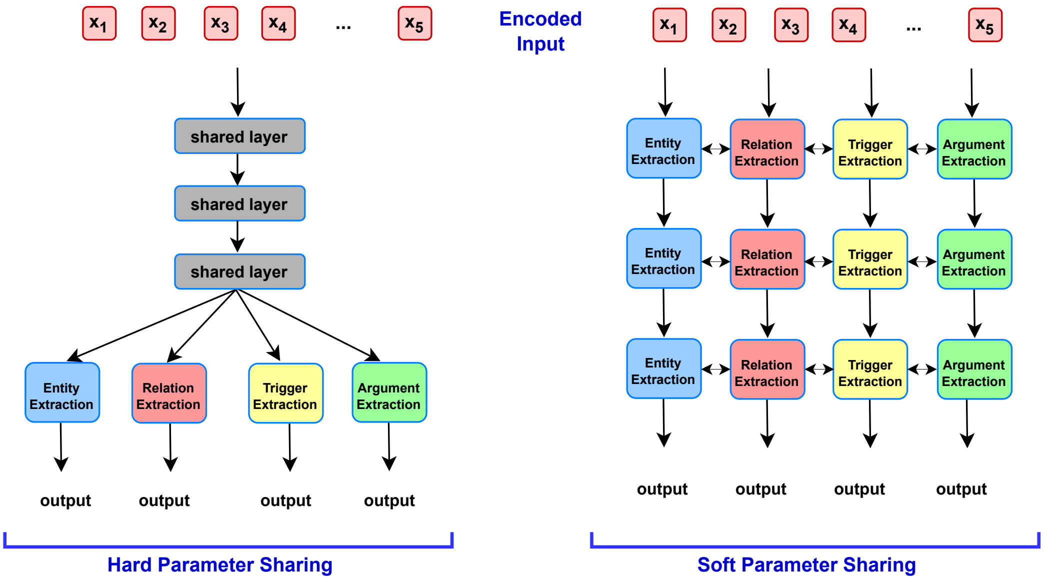


Figure 2.5: Comparison of Hard vs. Soft Parameter Sharing.

models, including multi-head selection approaches (Bekoulis et al., 2018), span-based approaches (Dixit and Al-Onaizan, 2019), sparse parameter sharing approaches (Chen et al., 2021a), and question-based object extraction approaches (Zhao et al., 2021). Models using parameter sharing can improve performance by learning shared features across tasks and reducing the overall number of parameters to be learned. However, finding the optimal level of sharing remains a challenge. Excessive sharing might restrict task-specific learning, while insufficient sharing might not fully exploit the benefits of multi-task learning.

In our information extraction models (cf. Chapters 3, 4, and 5), we prioritize flexibility and adaptability to handle the diverse requirements of each subtask. To achieve this, we avoid employing hard parameter sharing, since it can limit the model's ability to specialize for each task. Instead, we use shared representations, often from PLMs like BERT, serving as a powerful starting point for all subtasks. The use of these shared representations can be considered a form of parameter sharing. While the parameters themselves might not be directly shared across all subtasks, the pretrained knowledge embedded within these representations influences all downstream tasks. Moreover, we leverage a form of soft parameter sharing through joint training. Our information extraction modules are optimized using a joint loss function (cf. Sections 3.2 and 5.3), encouraging them to collaborate and share knowledge throughout the training process.

3. **Graph-based Models**:

   Another common used type of architecture is Graph Neural Networks (GNNs). GNNs employ message-passing techniques, where information iteratively flows between nodes, allowing the model to understand how connected entities and events influence each other. Additionally, attention mechanisms can be integrated within GNNs to focus on the most relevant parts of the graph for a specific entity or event, further enhancing the model's ability to make informed predictions.

Examples of related work for joint entity, relation, and event extraction include the framework of Luan et al. (2019) that constructs dynamic "span graphs" to capture relationships between entities. High-confidence entities are nodes, connected by edges representing relations or coreference. Edge weights based on confidence scores propagate information, refining entity representations within the graph. Another interesting work is that of (Lin et al., 2020), although they don't explicitly represent information as a graph structure like other graph-based models. It can be considered a conceptually graph-based approach due to the final stage of searching for the globally optimal extracted information as a unified graph, considering both local context (within a sentence) and global context (across sentences). As an extension to Lin et al. (2020), the work of Zhang and Ji (2021) leverages an Abstract Meaning Representation (AMR) parser to construct the information extraction graph and then uses two graph-based components: an aggregator to gather information from neighboring concepts and a decoder to extract knowledge elements based on the AMR graph structure. (Nguyen et al., 2021a) leverages a Graph Convolutional Network (GCN) to process a preliminary constructed information extraction graph and improve the representation of each element by incorporating information from its connected neighbors, explicitly capturing inter-dependencies between tasks.

4. **Autoregressive Graph Generation Frameworks**:

   This approach generates the output labels sequentially, one label at a time. At each step, the model predicts a label, i.e., entity, relation, and event types, for the current word based on previously predicted labels and the word's embedding. This sequential prediction allows autoregressive models to capture long-range dependencies in text, crucial for joint information extraction tasks like relation extraction and event argument extraction where entities and events can be far apart in a sentence. Popular architectures include:

   - **Transformers** (Vaswani et al., 2017): This powerful architecture has become a leading choice for joint information extraction due to its effectiveness in capturing long-range dependencies. Transformers use an encoder-decoder structure. The encoder processes the entire sentence, capturing relationships between words. The decoder then generates label predictions sequentially, attending to relevant parts of the encoded representation and previously predicted labels. Recent work by Li et al. (2020) leverages transformers for joint information extraction, proposing a Pointer-Generator Network that combines the strengths of both attention and copying mechanisms during label prediction. Additionally, Seo et al. (2020) introduce a Biaffine Dependency Parsing model within a transformer framework for joint entity and relation extraction, achieving state-of-the-art performance.

   - **Recurrent Neural Networks**: While transformers are currently dominant, RNNs, particularly Long Short-Term Memorys (LSTMs) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Cho et al., 2014b), have a history of success in joint information extraction. Their ability to handle long-range dependencies makes them suitable for this task. However, they can struggle with vanishing or exploding gradients during training compared to transformers.

- **Conditional Random Fields with Sequential Decoding** (Lafferty et al., 2001): While not strictly autoregressive in the same way as RNNs and transformers, CRFs with sequential decoding can be categorized here. Unlike traditional CRFs that predict all labels simultaneously, sequential decoding in CRFs allows for label prediction one word at a time, similar to RNNs and transformers. This enables CRFs to incorporate context from previous predictions during decoding.

Several studies have explored autoregressive frameworks for joint information extraction, such as chain-based recurrent neural networks for relation extraction (Ebrahimi and Dou, 2015). Miwa and Bansal (2016) introduce an entity and relation extraction approach using LSTMs with sequential and tree-structured representations. Gupta et al. (2016) explore using recurrent neural networks for joint named entity recognition and relation extraction in the context of table-filling tasks. Wu et al. (2017) investigate adversarial training for relation extraction. Nguyen et al. (2016) present a joint event extraction model using recurrent neural networks. Despite their success, these models suffer from some challenges such as:

- **Exposure Bias**: Autoregressive models can suffer from exposure bias, where the discrepancy between training and decoding stages leads to error propagation. During training, the model learns to predict labels using ground truth labels from the dataset. However, during decoding, which is the process of generating the output labels sequentially, the model must rely on its own previously predicted (potentially incorrect) labels. This inconsistency can cause errors to propagate through the sequence, negatively impacting overall performance.
- **Computational Cost**: Training and inference can be computationally expensive due to the sequential nature of prediction.

In our work, we aim to explore different techniques for modeling the interdependence between information extraction tasks (Chapter 5), particularly focusing on (1) developing hybrid models that leverage the strengths of both graph-based and autoregressive approaches for more robust joint information extraction, and (2) enhancing autoregressive frameworks to address exposure bias and improve computational efficiency.

2.2.4 Decoding Strategies in IE

While our research focuses on **novel techniques for Arabic information extraction** and **modeling interdependence in joint extraction tasks**, understanding decoding strategies remains crucial. These strategies, often employed with autoregressive models, determine how models generate the final output sequence based on the input text. In information extraction, this translates to selecting the most probable sequence of entities, relations, and event types. Here, we discuss some commonly used decoding strategies that can be applied within information extraction frameworks:

1. **Greedy Decoding**: Greedy decoding is a straightforward and efficient approach where the model predicts the output sequence one token at a time. At each step, it selects the token with the highest probability, aiming to build the most likely sequence incrementally. However, greedy decoding can get stuck in local optima, leading to suboptimal solutions in tasks requiring global context, like information extraction. Greedy decoding can be formulated as follows: Let $P(y_{t+1}|y_{1:t}, x)$ be the probability of the next token given the input sequence x and the generated sequence $y_{1:t}$. At each time step, the greedy decoding algorithm selects the token with the highest probability:

$$y_{t+1} = \text{argmax}_y P(y_{t+1}|y_{1:t}, x)$$

   where y_{t+1} is the selected token at time step $t + 1$.

2. **Beam Search**: Beam search is a heuristic search algorithm used to find the most likely sequence of output tokens. It maintains a fixed-size set of candidate sequences (beam width) at each decoding step and expands them based on the probabilities of next tokens. Beam search allows the model to explore multiple hypotheses simultaneously, improving the quality of generated sequences compared to greedy decoding. The beam search algorithm can be formulated as follows: Let B_t be the set of candidate sequences at time step t, and $P(y_{t+1}|y_{1:t}, x)$ be the probability of the next token given the input sequence x and the generated sequence $y_{1:t}$. At each time step, the beam search algorithm selects the k most probable candidate sequences based on their scores:

$$B_{t+1} = \text{Top-K}(B_t \times P(y_{t+1}|y_{1:t}, x), k)$$

   where Top-K(S, k) selects the k elements with the highest scores from set S.

3. **Viterbi Decoding** (Viterbi, 1967): This method is commonly used in sequence labeling tasks such as part-of-speech tagging and named entity recognition. Viterbi decoding uses dynamic programming to find the most probable sequence of output labels given the input sequence. Viterbi decoding can be formulated using the Viterbi algorithm, which recursively computes the highest probability path to each state at each time step. Let $v_t(j)$ denote the probability of the most probable sequence of length t ending in state j. The Viterbi algorithm computes $v_t(j)$ as:

$$v_t(j) = \max_i [v_{t-1}(i) \times a_{ij} \times b_j(x_t)]$$

   where a_{ij} is the transition probability from state i to state j, $b_j(x_t)$ is the emission probability of observing symbol x_t in state j, and i ranges over all states at time step $t - 1$.

Even though our primary focus lies on novel techniques for modeling interdependence in joint information extraction, understanding decoding strategies offers several advantages for our work: (1) By analyzing the strengths and weaknesses of different decoding strategies, we can make informed decisions. We can select or adapt existing autroregressive models for our joint information extraction framework while considering the specific decoding approach they employ. (2) Although not our central focus, knowledge of decoding strategies allows us to combine these with our proposed techniques for interdependence modeling to achieve even better results in joint information extraction. This could potentially address challenges like exposure bias and improve the quality of generated information extractions.

2.2.5 Arabic NLP and Information Extraction

Arabic NLP Initiatives A lot of efforts have been made to advance Arabic NLP research. This is manifested by many Arabic NLP workshops and conferences. Notable examples include the International Symposium on Computer and Arabic Language (ISCAL) editions in 2009 and 2007, and the workshop on Computational Approaches to Arabic Script-based Languages. Additionally, WANLP, The Arabic Natural Language Processing Workshop, has seen seven editions, hosted at prominent conferences such as EMNLP, ACL, COLING, and EACL, covering years from 2014 to 2022. More recently, ArabicNLP 2023 continued this trajectory, further solidifying the commitment to advancing Arabic NLP research.

Chronology of Arabic Information Extraction The development of Arabic information extraction mirrors the broader field, transitioning from rule-based approaches to machine learning and, more recently, deep learning techniques. In the early stages, before the dominance of machine learning, rule-based systems reigned supreme. Researchers focused on crafting intricate rules to identify and extract entities and relationships within Arabic text. These include Arabic name recognizers leveraging pattern matching and morphological analysis (Maloney and Niv, 1998). NERA is one of the early rule-based systems known developed for Arabic using dictionaries and regular expressions to extract 10 key named entity types (Shaalan and Raza, 2007, 2008, 2009). Abdallah et al. (2012) improved the results of NERA by combining machine learning and rule-based system. Other work (Traboulsi, 2009) leveraged local grammar to extract person names, other work (Elsebai et al., 2009) leveraged morphological analyzers and keyword-guided phrase matching. For relations, Ben Hamadou et al. (2010) presents a rule-based system to extract functional relations from Arabic text. Al Zamil and Al-Radaideh (2014) present a pattern-based system that uses a seed ontology to automatically extract antonym relationships from Arabic text.

As computational power increased, statistical methods emerged, leveraging machine learning algorithms to learn patterns from labeled data. In this context, a lot of work has been done on entity and relation extraction, exploring different architectures such as SVMs Benajiba et al. (2008b); Falih and Omar (2015); Hamad and Abushaala (2023), CRFs Benajiba and Rosso (2008); Abdul-Hamid and Darwish (2010); Alzboun et al. (2018); Hudhud et al. (2021), and MEMs Benajiba et al. (2007); Benajiba and Rosso (2007).

Recognizing the strengths of both approaches, some researchers investigated hybrid systems that combine rule-based and machine learning techniques (Oudah and Shaalan, 2012; Abdallah et al., 2012; Koulali and Meziane, 2012). Finally, the recent surge in deep learning had a significant impact on Arabic information extraction. Deep learning architectures like BiRNNs (Ali et al., 2018) and CNNs (Benajiba et al., 2010) have demonstrated promising results in named entity recognition tasks.

Beyond Supervised Learning In addition to supervised learning methods, various techniques have been explored for Arabic relation extraction:

- **Distant Supervision**: This technique leverages readily available resources like knowledge bases to automatically generate training data for relation extraction (Mohamed et al., 2015). The underlying assumption is that if two entities are linked in a knowledge base with a specific relation, then this relation likely holds between mentions of these entities in text. While convenient, distant supervision can introduce noise due to potential inaccuracies in knowledge bases.

- **Semi-Supervised Learning**: This approach uses a combination of labeled and unlabeled data to train a relation extraction model (Sarhan et al., 2016). Techniques like pattern bootstrapping can be employed to iteratively improve the model. Starting with a small set of labeled data, the model can be used to identify potential relation patterns in unlabeled data. These patterns can then be reviewed by human experts for verification and used to further train the model. This approach can be particularly useful when labeled data is scarce.

- **Cross-Lingual Learning**: This technique leverages resources from languages with more abundant labeled data, like English, to improve relation extraction in languages like Arabic (Taghizadeh et al., 2018; Subburathinam et al., 2019; Nguyen et al., 2021b). By transferring knowledge learned from a related language, cross-lingual models can potentially improve performance in resource-scarce languages. This approach often relies on techniques like multilingual embeddings that capture semantic similarities between words across languages.

Limited Research in Event Extraction While event extraction is a valuable field of study, research specifically focused on Arabic text remains limited. Existing work has primarily addressed event extraction from social media data. Here are some examples:

- Focusing on Arabic tweets, AL-Smadi and Qawasmeh (2016) proposed a knowledge-based approach for event extraction.

- Alsaedi and Burnap (2015) presented a clustering-based framework to detect real-world events from Twitter data in Arabic.

- Harrag and Gueliani (2020) built a system that uses recurrent neural networks to extract food hazard events from social media.

- Alsaedi and Burnap (2015) proposed a method that combines clustering and Naive Bayes to identify disruptive events from Arabic social media posts on Twitter.

Beyond social media, Ahmad et al. (2021) developed a Graph Attention Transformer Encoder for cross-lingual relation and event extraction, achieving promising results in structured contextual representation generation. For standard Arabic text, Baradaran and Minaei-Bidgoli (2015) compared three classification methods for event trigger and argument extraction. Their findings suggest that rule-based and SVM-based data-oriented approaches outperform the semantic-oriented approach relying on lexical chains.

Our goal is to enhance language technologies dedicated for Arabic and pave the way for further advancements in information extraction from Arabic text data. To achieve this, our work in Chapters 3 and 4 advances the state-of-the-art of the field. While previous research has primarily concentrated on named entity recognition and relation extraction, our contributions extend beyond these domains to encompass event extraction, filling a crucial gap in Arabic information extraction research. Additionally, our exploration of cross-dialectal named entity recognition in Arabic tackles the challenges posed by dialectal diversity and the scarcity of annotated resources, offering innovative solutions to address these issues.

2.3 Datasets for Information Extraction

Information extraction relies heavily on well-annotated datasets for training and evaluating models. Some research communities organize shared tasks and workshops that significantly contribute to the development of information extraction techniques. These conferences and workshops provide crucial benchmarks and datasets for evaluating the performance of different approaches. Here are some prominent examples:

- **Automatic Content Extraction (ACE)** (Doddington et al., 2004): The ACE corpus is a widely used dataset for information extraction tasks. It comprises data in English, Chinese, and Arabic from sources like broadcast transcripts, newswires, and newspapers. The corpus offers separate training and testing data, with a defined vocabulary of entities, mentions, and relations between them. ACE tasks include entity detection and tracking, relation detection and characterization, event detection and characterization, entity linking, and timestamp extraction.

- **Conference on Computational Natural Language Learning (CoNLL)**: The CoNLL conferences focus on natural language understanding tasks, including named entity recognition. The CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) dataset provides a benchmark for evaluating NER systems on newswire text.

- **Message Understanding Conference (MUC)** (Grishman and Sundheim, 1996): The MUC conferences from 1987 to 1997 focused on tasks related to information extraction, including named entity recognition, relation extraction, and event detection. The MUC datasets, notably MUC-3 and MUC-4 which are publicly available[3], played a crucial role in the early development of these techniques.

[3] https://www-nlpir.nist.gov/related_projects/muc/muc_data/muc_data_index.html

- **Semantic Evaluation (Sem-Eval)**[4]: This yearly workshop offers various datasets for tasks like relation extraction, often focusing on specific domains like scientific articles.
- **OntoNotes** (Hovy et al., 2006): Developed collaboratively across various US institutes, OntoNotes provides a large, human-annotated corpus encompassing diverse textual genres, such as telephone speech, broadcast news, etc., in multiple languages. It's a widely used benchmark for named entity recognition tasks, with each release building upon previous versions and encompassing various data sources. OntoNotes offers a rich and challenging dataset for NER evaluation, containing around 2.945 million tokens.

Beyond the conference/community-driven categorization, information extraction datasets can be categorized based on various other criteria, each offering insights into their suitability for specific tasks. Here's an overview of common categorization methods:

- **Task Focus**: This categorization groups datasets based on the specific information extraction task they support:
  - **Entity Extraction**: These datasets focus on identifying and classifying entities within text. Examples include MUC-7 (Chinchor, 2001), CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003), and WiNER (Ghaddar and Langlais, 2017).
  - **Relation Extraction**: These datasets focus on identifying relationships between previously identified entities. Examples include CoNLL 2004 (Roth and Yih, 2004), ChemProt, and NYT for The New York Times annotated corpus (Sandhaus, 2008).
  - **Event Extraction**: These datasets focus on identifying and characterizing events mentioned within text data, including event types, arguments, and temporal information. Examples include the ACE 2005 Event Corpus Walker and Consortium (2005), and GENIA (Kim et al., 2003).
- **Data Source**: This categorization groups datasets based on the origin of the text data they contain:
  - **News Articles**: Offer a formal style and focus on current events. Examples include MUC-7 (MUC, 1998), and the ACE 2005 Event Corpus (Walker and Consortium, 2005).
  - **Social Media**: Offer informal language and user-generated content. May require specific preprocessing techniques. Examples include Twitter Event Detection datasets (Zubiaga, 2018).
  - **Web Documents**: Can encompass a wide variety of text types and styles, requiring broader adaptation strategies for IE models. Examples include WiNER (Ghaddar and Langlais, 2017).
- **Language**: This categorization groups datasets based on the language they represent:

[4] https://dblp.org/db/conf/semeval/index.html

  - **English**: The most common language for information extraction datasets, offering a wider selection for training and evaluation. Examples include MUC-7, CoNLL 2003, and the ACE 2005 Event Corpus.
  - **Multilingual Datasets**: As research expands beyond English, datasets in various languages are becoming increasingly important. Examples include CoNLL 2003, and Wikipedia NER.

- **Domain Specificity**: This categorization groups datasets based on the specific domain or field they represent:
  - **General Domain**: Applicable to a broad range of text data and topics. Examples: MUC-7, CoNLL 2003
  - **Biomedical Domain**: Focuses on medical text and terminology, requiring domain-specific knowledge for accurate extraction. Examples include Genia.
  - **Financial Domain**: Focuses on financial news and reports, requiring an understanding of financial terms and entities. Examples include ChFinAnn (Zheng et al., 2019).

In this thesis, we focus primarily on the general domain of information extraction. This focus allows us to explore and advance methodologies applicable across diverse contexts. To achieve this, we have selected two prominent datasets, notably CoNLL04 and ACE05. The CoNLL04 dataset presents a comprehensive set of challenges in entity and relation extraction within the English language domain. This established benchmark provides a robust foundation for our experimental investigations. The ACE05, dataset on the other hand, extends the scope by encompassing entity, relation, and event extraction tasks across multiple languages, including Arabic. This broader dataset allows us to explore the generalizability of our approaches and their potential for adaptation to different languages. By leveraging these datasets, we ensure relevance of our experimental evaluations, and align ourselves with established benchmarks in the field, facilitating meaningful comparisons and contributions to the information extraction field. In the following subsections, we provide detailed insights into these two datasets.

The ACE05 Dataset The ACE 2005 multilingual training corpus (ACE05) is a reference resource for research and development in multilingual information extraction tasks. It provides annotated text data in English, Arabic, Spanish, and Chinese, including diverse sources like news articles, newswires, and online text. ACE05 offers rich annotations for various information extraction tasks:

- Entities: Categorized into 7 types and 45 subtypes as detailed in Table 2.1.
- Relations: Categorized into 6 types and 18 subtypes as detailed in Table 2.3.
- Event triggers: Categorized into 8 types and 33 subtypes as detailed in Table 2.2.

Type	**Subtypes**
Person (PER)	Individual, Group, Indefinite
Organization (ORG)	Government, Commercial, Educational, Entertainment, Non-Governmental Organizations, Media, Religious, Medical-Science, Sports
Geo-political Entity (GPE)	Continent, Nation, State-or-Province, County-or-District, Population-Center, GPE-Cluster, Special
Location (LOC)	Address, Boundary, Celestial, Water-Body, Land-Region-natural, Region-International, Region-General.
Facility (FAC)	Airport, Plant, Building-or-Grounds, Subarea-Facility, Path.
Vehicle (VEH)	Air, Land, Water, Subarea-Vehicle, Underspecified.
Weapon (WEA)	Blunt, Exploding, Sharp, Chemical, Biological, Shooting, Projectile, Nuclear, Underspecified.

Table 2.1: ACE05 Entity Types and Subtypes.

Additionally, ACE05 encompasses 22 event argument types (e.g., `Agent`, `Artifact`). These solely focus on the type of entity involved in an event, distinguishing them from the other categories that include subtypes.

There are constraints on the types of entities that can participate in each relation, depending on its type. These constraints ensure logical and contextually appropriate connections between entities, as detailed in Table 2.3. Similarly, each event type can have predefined event roles, which themselves must correspond to specific predefined entity types. For example, a `Marry` event type can have roles such as person (the people who are married), time (when the marriage takes place), and place (where the marriage takes place). For person, the entity type must be `Person` (`PER`). For time, the entity type must be `Time` (`TIME`). For place, the entity types can be `Geo-Political Entity` (`GPE`), `Location` (`LOC`), or `Facility` (`FAC`). However, due to the extensive number of event roles, we do not present these constraints here. Thus, for comprehensive and complete guidelines on entity, relation, and event annotations, please refer to the ACE 2005 annotation tasks and specifications available at LDC Annotation Tasks and Specifications.

The CoNLL04 Dataset The CoNLL-2004 (CoNLL04) Shared Task dataset is a benchmark widely used for evaluating joint entity and relation extraction methods. CoNLL04 defines the four main entity types presented in Table 2.4, and five types of relations presented in Table 2.5.

Type	Subtypes
Life	Be-born, Marry, Divorce, Injure, Die.
Movement	Transport.
Transaction	Transfer-ownership, Transfer-money.
Business	Start-org, Merge-org, Declare-bankruptcy, End-org.
Conflict	Attack, Demonstrate.
Contact	Meet, Phone-write.
Personnel	Start-position, End-position, Nominate, Elect.
Justice	Arrest-jail, Release-parole, Trial-hearing, Charge-indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon.

Table 2.2: ACE05 Event Trigger Types and Subtypes.

2.4 Evaluation Metrics

Evaluating the performance of information extraction models is crucial for assessing their effectiveness and comparing different approaches. This section explores some commonly used metrics for evaluating various information extraction tasks. Core evaluation metrics include:

- **Precision**: Measures the proportion of predicted labels that are actually correct. It is calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.1}$$

  Where TP denotes True Positives, the number of correctly predicted labels, and FP denotes False Positives, the number of labels incorrectly predicted as positive.

- **Recall**: Measures the proportion of actual positive labels that are correctly identified by the model. It is calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.2}$$

  Where FN denotes False Negatives, the number of actual positive labels that the model missed.

- **F1 score**: Combines precision and recall into a single metric, providing a balanced view of model performance. It is calculated as:

$$\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.3}$$

Type	Subtypes	Arg1	Arg2
Physical (PHYS)	Located Near	PER PER, FAC, GPE, LOC	FAC, LOC, GPE FAC, GPE, LOC
Part-whole (PART-WHOLE)	Geographical Subsidiary Artifact	FAC, LOC, GPE ORG VEH	FAC, LOC, GPE ORG, GPE VEH, WEA
Personal-social (PER-SOC)	Business Family Lasting-personal	PER PER PER	PER PER PER
ORG-Affiliation (ORG-AFF)	Employment Ownership Founder Student-Alum Sports-Affiliation Investor-Shareholder Membership	PER PER PER, ORG PER PER PER, ORG, GPE PER, ORG, GPE	ORG, GPE ORG ORG, GPE ORG ORG ORG, GPE ORG
Agent-Artifact (ART)	User-Owner-Inventor-Manufacturer	PER, ORG, GPE	WEA, VEH, FAC
Gen-Affiliation (GEN-AFF)	Citizen-Resident-Religion-Ethnicity Org-Location-Origin	PER ORG	PER, LOC, GPE, ORG LOC, GPE

Table 2.3: ACE05 Relation Types and Subtypes.

Type	Definition
People (Peop)	Represents individuals and pronouns referring to them.
Organization (Org)	Encompasses companies, institutions, and other established entities.
Location (Loc)	Includes geographical entities like countries, cities, and landmarks.
Other (Other)	Covers other relevant entities, such as dates, monetary values, and percentages.

Table 2.4: CoNLL04 Entity Types and Subtypes.

Both entity and event trigger extraction tasks commonly use the F1 score for evaluation. However, relation extraction tasks require a nuanced evaluation approach due to the complexity of predicting entity boundaries and relation types. The evaluation of relation

Type	Definition
Work_For	Captures the employment relationship between a person and an organization.
OrgBased_In	Indicates the location where an organization is headquartered or operates.
Live_In	Represents the place of residence for a person.
Kill	Identifies acts of violence resulting in one person's death by another.
Located_In	Specifies a geographical containment, where one location is situated within another.

Table 2.5: CoNLL04 Relation Types and Subtypes.

extraction can involve the following techniques:

- **Strict Evaluation**: This method requires both the correct prediction of entity boundaries and their types, along with the accurate prediction of the relation type between entities.
- **Relaxed Evaluation**: In this approach, the focus is on correctly predicting entity boundaries and the relation type, without strict adherence to exact entity types.

Moreover, relation extraction evaluation can include directed and non-directed assessments, which determine whether the predicted order of the two entities is significant.

In event argument extraction evaluation, the assessment comprises two primary aspects: the accurate identification of the event trigger and its corresponding type, and the precise identification of the event argument along with its associated type.

2.5 Conclusion

In this chapter, we reviewed the core tasks of information extraction (Section 2.1), including entity, relation, event trigger, and event argument extraction. Using ACE program definitions ensured clarity and will facilitate comparisons with existing research in subsequent chapters.

We discussed various learning paradigms in Subsection 2.2.1 such as supervised, unsupervised, reinforcement, and transfer learning, highlighting their contributions and limitations. We emphasized that we will focus on supervised learning in all the following chapters. The historical evolution of information extraction (Subsection 2.2.2) demonstrated a shift from rule-based systems to machine learning and deep learning models, each enhancing the ability to handle complex text patterns.

We also reviewed joint information extraction (Subsection 2.2.3), exploring parameter sharing, graph-based models, and autoregressive frameworks to address limitations like error propagation and limited information sharing. We will adopt joint multitask learning in our work addressing the four tasks of information extraction (Chapters 3 and 5). Additionally, we presented decoding strategies (Subsection 2.2.4) used in information extraction, noting that we will use greedy decoding in both Chapters 3 and 4, and beam search in Chapter 5.

We then covered specific challenges and advancements in Arabic information extraction (Subsection 2.2.5), most of which we will address in Chapters 3 and 4.

We reviewed available information extraction datasets (Section 2.3), highlighting key benchmarks like CoNLL04 and ACE05, which we will use to train and evaluate our models.

Lastly, we outlined evaluation metrics (Section 2.4) such as precision, recall, and F1 scores, with additional techniques for relation extraction and event argument extraction, providing a framework for assessing model performance.

This review sets the stage for the subsequent chapters, where we present our proposed models and techniques for advancing Arabic information extraction and joint information extraction tasks.

Chapter 3

Joint Entity, Relation, and Event Extraction for Arabic

3.1 Introduction

With Arabic being the world's fifth most spoken language, and the language of regions rich in culture and current events, extracting valuable information from Arabic texts holds significant potential across a range of domains. While information extraction has seen significant advancements for languages like English, Chinese, and Spanish, applying these techniques to Arabic presents unique challenges. In this work we will only focus on Modern Standard Arabic (MSA).

Challenges of Arabic Information Extraction

- **Complex Morphology**: Unlike English and languages with clear word boundaries, MSA morphology is non-concatenative. Words are formed by attaching prefixes, suffixes, and roots, often blurring the lines between word and morpheme. Entities can span across these units, making it difficult to segment text for traditional IE methods.
- **Rich Vocabulary and Diacritics**: MSA has a rich vocabulary with many words having multiple meanings depending on context or on the corresponding diacritization. This ambiguity can lead to misidentification of entities and event arguments.
- **Lack of Resources**: Compared to well-resourced languages like English, MSA has limited publicly available annotated data for training IE models. This scarcity hinders the development of robust Arabic IE systems.

Previous Work Significant research has been conducted in Arabic NER, with notable contributions from studies like (Benajiba et al., 2004; Oudah and Shaalan, 2012; Naji, 2012). However, efforts in Relation Extraction (RE) and Event Detection (ED) for Arabic have been limited (Taghizadeh et al., 2018; AL-Smadi and Qawasmeh, 2016). Notably, there is a complete absence of prior work addressing these tasks jointly. This chapter aims to bridge this gap by introducing a novel joint multi-tasking system for four crucial Arabic IE tasks: entity, relation, event trigger, and event argument extraction.

Our Proposed Model Our model builds upon the graph-based approach developed by Lin et al. (2020). It operates in two stages:

- **Identification Stage**: This stage employs two CRFs (Lafferty et al., 2001) with BIO-based tags to identify entities and event triggers, essentially creating the nodes of our information graph.
- **Information Graph Construction**: A greedy decoding strategy constructs the information graph depicting the relationships between identified entities and event triggers (edges of the graph).

Addressing the Challenges Arabic's rich morphology necessitates modeling at a subword level to accurately capture and represent entities. Thus, we explore two subword modeling approaches:

- **Word Tokenization**: As a preprocessing step, we split morphologically complex words into tokens, each of which corresponds to (or is a part of) one entity at most. An entity can thus be modeled as a sequence of tokens using the standard BIO tags.
- **Augmented BIO Tags**: We modify the standard BIO scheme to encode multiple entities within a single word, eliminating the need for prior tokenization.

A detailed comparison of these approaches and their impact on IE performance is presented in Section 3.3.

Contributions This work makes the following key contributions:

- **First neural joint IE model for Arabic:** We present ArabIE (§3.2), the first neural joint IE model for Arabic. It tackles four essential tasks simultaneously: Named Entity Recognition, Relation Extraction, Event Recognition, and Argument Detection. On the ACE 2005 benchmark dataset, ArabIE achieves state-of-the-art results. We show that its performance is comparable to existing top models for other languages (Section 3.5), highlighting its potential for broader impact.
- **Empirical study of tokenization and IE performance**: We conduct a comprehensive empirical study (Section 3.3) to analyze the complex interplay between tokenization strategies and IE performance.
- **Error Analysis**: We perform a detailed error analysis to identify the specific weaknesses and limitations of our model, providing insights for future improvements and research directions.

3.2 Model Architecture

Given a text document as input, we aim at extracting, from each sentence, entities and binary relations between them, event triggers, and their arguments. Formally, for an input sequence $\boldsymbol{x}$ of length L , the information extraction task is the operation that yields, as an output, a graph $G = (V, E)$ whose nodes V are spans of tokens of the input sequence representing identified entities and triggers, and whose edges E represent relations between two entities or event roles (relations between event triggers and their arguments entities). Each node and edge in the graph has a type. Similar to Lin et al. (2020), our model performs end-to-end IE in four stages.

3.2.1 Token encoding

We explore various combinations of representations from different layers of BERT to encode the input sequence, inspired by token encodings of Lin et al. (2020) for English data. After thorough experimentation, we find that concatenating the output from BERT's last and third-to-last layers yields optimal performance across most subtasks. This choice is motivated by the findings of Jawahar et al. (2019), who demonstrated that the last layers of BERT capture rich semantic information crucial for processing Arabic texts. Before encoding, input sequences may undergo additional (optional) tokenization as part of a preprocessing step (see §3.3). The embedding for the i-th token $\boldsymbol{x}_i$ of the input sequence $\boldsymbol{x}$ is computed as follows:

$$\boldsymbol{x}_i = [\mathbf{BERT}_i^{N-1}; \mathbf{BERT}_i^{N-3}]$$

Here, $\mathbf{BERT}_i^j$ represents the output vector for the i-th token from the j-th layer of the BERT model, with N denoting the total number of layers.

3.2.2 Identification

Token embeddings are passed to a network composed of a Feed-Forward Network (FFN) followed by a Conditional Random Field (CRF). The network leverages the BIO scheme to identify spans of tokens corresponding to entities or event triggers within the sequence.

The FFN The FFN takes the token embedding $\boldsymbol{x}_i$ of each token i in the sequence $\boldsymbol{x}$ as input and computes a score vector $\boldsymbol{y}_i$ from the BIO tag set (`B-PER, I-PER, B-LOC, I-LOC, ..., O`):

$$\boldsymbol{y}_i = \text{FFN}(\boldsymbol{x}_i)$$

Separate CRFs for Entities and Triggers We then employ two separate CRF models, one for entity and the other for triggers, allowing each model to specialize in its respective task. Each CRF predicts a sequence of labels using the BIO scheme for the corresponding element type.

Score for a Tag Path The score $s(x, z)$ for a tag path $\boldsymbol{z} = (\boldsymbol{z}_1, ..., \boldsymbol{z}_L)$ is calculated as:

$$s(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{L} \boldsymbol{y}_i \cdot \boldsymbol{z}_i + \sum_{i=1}^{L+1} A_{\boldsymbol{z}_{i-1}, \boldsymbol{z}_i}$$

Here, A is the transition matrix inferred by the CRF capturing dependencies between labels, and $\boldsymbol{z}_i$ is the label predicted by the CRF for the i-th token (BIO scheme). Note that the left term from the addition reflects the compatibility between the FFN output $\boldsymbol{y}_i$ and the predicted entity or trigger label $\boldsymbol{z}_i$, and the second term captures the transition score between consecutive entity or trigger labels.

Labeling and Overlap The predicted label sequence from the CRFs segments the input sequence. This ensures that identified entities and triggers won't overlap within their respective categories (entities or triggers). However, entities and triggers can coexist within the same token in some cases. For example, in the verb "أوقفت" (*Awqft; she arrested)*, we can distinguish the following annotated information:

1. The whole verb "أوقفت" is a trigger of type `Justice`;

2. The pronoun "ت" (*t*) is an entity of type `Person`.

3.2.3 Classification

At this stage, entities and triggers are identified, but their types are not yet assigned. A fixed-size representation for each span v_i is computed as the average of its word representations.

The output is passed to an FFN to obtain a score for each possible type. Again, we use separate FFNs for entities and triggers.

Scoring relations and event roles is performed in a similar manner. An edge between two spans is represented by concatenating their vectors. A relation edge links two entities while a role edge links a trigger to an entity. Representations of edges are passed to an FFN to compute a score for each relation or role type. A special *none* label to indicate the absence thereof. We also use a separate FFN for relations and roles.

3.2.4 Decoding

To construct the final information graph representing the extracted entities, triggers, relations, and event arguments, we use an unconstrained greedy decoding approach. For each node (identified entity/trigger) and edge (relationship between nodes) in the potential graph, we select the label type with the highest predicted score from the classification stage. This iterative process continues until all nodes and edges have assigned types, resulting in the final information graph.

We explored incorporating a penalty term on invalid graph configurations into the overall score and decoding with beam search as done by Lin et al. (2020). However, in our experiments, neither of these approaches yielded significant improvements over the basic unconstrained greedy decoding.

3.2.5 Training

We employ joint end-to-end training to optimize the parameters of all networks simultaneously. The training objective is to minimize the sum of the individual loss functions associated with each subtask: entity/trigger identification (CRFs) and classification (FFNs) for entities, triggers, relations, and event arguments.

For the CRF layers, we use the negative log-likelihood of the gold BIO paths as the loss function, denoted as $\mathcal{L}_{\text{CRF}}$:

$$\mathcal{L}_{\text{CRF}} = -\log p(\boldsymbol{z}^*|x) = -(s(\boldsymbol{x}, \boldsymbol{z}^*) - \log \sum_{\boldsymbol{z}' \in Z} e^{s(\boldsymbol{x}, \boldsymbol{z}')})$$

Here, Z is the set of all possible tag paths for a given sentence.

For the FFN classifiers used in the classification stage of all tasks (entity, trigger, relations, and event arguments types), we also employ the negative log-likelihood loss function, denoted as $\mathcal{L}_{\text{FFN}^t}$:

$$\mathcal{L}_{\text{FFN}^t} = -\sum_{n=1}^{N_t} \boldsymbol{y}_i^{*t} \cdot \log(\boldsymbol{y}_i^t)$$

Here, $\mathcal{L}_{\text{FFN}^t}$ denotes the loss for task $t \in \{\text{entity}, \text{relation}, \text{trigger}, \text{role}\}$, $\boldsymbol{y}_i^t$ represents the predicted score vector for the i-th span or edge in task t, $\boldsymbol{y}_i^{*t}$ represents the corresponding gold-standard label vector, N_t represents the number of training instances for task t.

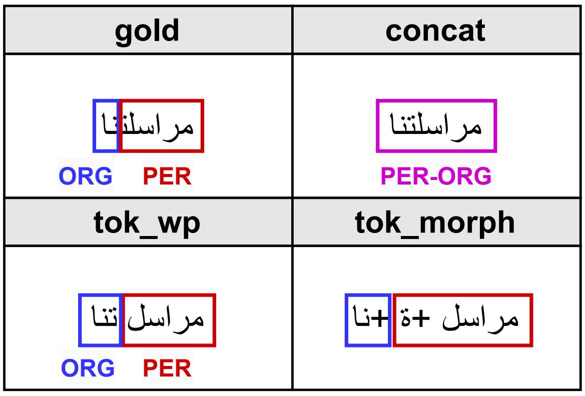


Figure 3.1: Examples of Different Arabic Tokenization Approaches.

3.3 Particularities of the Arabic Language

As discussed in Section 4.1 regarding Arabic's morphology, a word in Arabic can hold two or more entities anchored on its root or affixes. For example, consider the word "مراسلتنا" *(mrAsltnA; our reporter)*. This word contains two distinct entities: "مراسلة" *(mrAslp; reporter)* of type `Person` (`PER`) and نا *(nA; our)* of type `Organization` (`ORG`)[1]. This example cannot be handled by traditional sequence labeling approaches, which typically assign a single label to each token in the sequence. Previous NER systems using sequence labeling often treated such cases as anomalies and simply discarded subword entities (Benajiba et al., 2008a). However, this approach results in a loss of valuable information.

We propose two solutions to address this problem detailed in the following paragraphs.

Word tokenization Subword entities typically correspond to *morphemes*. We, therefore, use a morphological analyzer to tokenize words in context. The probability that each resulting token corresponds to multiple entities decreases dramatically. In practice, we use the analyzer provided by CamelTools (Obeid et al., 2020) and refer to this tokenization scheme by **tok_morph**.

For the example word "مراسلتنا", the morphological analyzer segments it into three morphemes: "مراسل +ة +نا" *(mrAsl +p +nA)*. The first two tokens correspond to the entity "*Reporter*" of type `Person`, while the third token "*Our*" corresponds to an entity of type `Organization`.

To create training data for the tokenized sequences, we align each word with its corresponding morphemes at the character level. This alignment is then used to project gold standard entities onto the resulting tokens. An entity is projected onto a token if the majority

[1] This example is taken from the ACE 2005 corpus. We use the Buckwalter (Buckwalter) transliteration scheme for Romanization. Note that the taa' marbuuTa (ة; p) transforms to taa' (ت; t) when attached to the suffix (نا; nA).

of its characters align with that token. However, if multiple entities are projected onto the same token, only one is chosen randomly.

To validate our hypotheses that morphemes are the right level for modeling entities, we compare the morphological analyzer to Word Pieces (Wu et al., 2016), a statistical tokenizer which does not necessarily produce valid affixes. This tokenizer produces“ مراسل تنا” *(mrAsl tnA)* for the example word where the second token is not a morphologically valid suffix and does not exactly match the gold entity “نا” *(nA; our)*. We refer to this tokenization scheme by **tok_wp**.

Data Loss Quantification Projection of entities onto tokens is not always perfect either because an entity doesn't correspond to a morpheme in gold data; the tokenizer doesn't produce a valid morpheme; or both. This results in some data loss that we quantify in Table 3.2 for the different tokenization schemes. This data loss is taken into account during the evaluation phase.

Consider the sentence “سألتها عن الجيران” (*s>lthA En AljyrAn; she asked her about the neighbours)*. Here, we have three entities:

- “ت” *(t)*: entity of type person;
- “ها” *(hA)*: entity of type person;
- “الجيران” *(AljyrAn)*: entity of type person.

There is also a relation of type personal-social (PER-SOC) between “ها” and “الجيران” . However, the **tok_wp** approach tokenizes the sentence as “سأل تها عن الجيران”. The problematic aspect is the token “تها”. It combines the two entities “ت” and “ها” but doesn't represent either one accurately. This makes it impossible to project both entities onto the single token. Therefore, we randomly choose one entity to project onto “تها”. If “ت” is chosen, then “ها” is discarded. This results in the loss of the entity “ها” and the PER-SOC relation between it and “الجيران”.

Augmented BIO Tags An alternative approach we explore is label concatenation. Here, instead of tokenizing words and projecting entities, we combine the labels of subword entities into a single*complex* entity. For example, the word “مراسلتنا” would be labeled as PER-ORG using this scheme.

This approach is appealing due to its simplicity but has limitations. Firstly, it significantly increases the size of the label set, as some words contain up to four entities. In practice, to mitigate this issue, we restrict the label set to the labels observed in the training data. This

Source	Files	Words	Entities	Relations	Triggers	Roles
NW	221	53026	17105	2674	1270	2957
BN	127	26907	9099	1606	870	1762
WL	55	20181	6234	439	130	256
Total	403	100114	32438	4719	2270	4975

Table 3.1: General Statistics of Raw Arabic ACE05.

ensures the model focuses on relevant entities encountered during training. We refer to this tokenization scheme by **concat**.

Figure 3.1 summarizes the different approaches adopted on the example of the word "مراسلتنا", where entities are framed in different colors w.r.t their label types.

3.4 Experimental Setup

In this section, we describe the dataset used for training and evaluation, the preprocessing steps applied, and the evaluation metrics.

3.4.1 Dataset and Preprocessing

We use the Arabic corpus ACE05 provided by the LDC[2], which consists of various document types annotated with entities, relations, and events. The corpus encompasses broadcast news, news wire, and weblog files, each annotated with rich information. Table 3.1 presents general statistics of the raw Arabic ACE05 data, illustrating the distribution of words, entities, relations, event triggers, and event arguments (roles) across different document sources.

Despite its availability since 2006, limited work has been conducted on ACE05 for entity extraction, and no efforts have been made towards relation or event extraction. These previous works are further discussed in detail in Section 3.7. To facilitate experimentation, we randomly split the ACE05 data into 80% training, 10% development, and 10% testing sets, as no official split is provided. Our splits are publicly available [3] for further developments.

Segmentation Document segmentation into sentences is performed using punctuation marks, except for the broadcast news subcorpus, which is segmented into fixed-length sentences due to the absence of punctuation. It's worth noting that document segmentation may result in the loss of some entities and triggers, along with their associated relations and roles, if a sentence boundary happens to occur within them. Comparing train rows of **gold** and **segm** in Table 3.2 allows to quantify the data loss after the segmentation phase.

[2] https://catalog.ldc.upenn.edu/LDC2006T06

[3] https://github.com/niamaelkhbir/ArabIE

Tokenization	Split	Entities	Relations	Triggers	Roles
gold	train	26178	3801	1831	3346
	dev	3296	508	235	418
	test	2946	400	204	352
segm	train	26065	3727	1831	3181
concat	train	26065	3727	1831	3181
tok_wp	train	25554	3416	1831	3176
tok_morph	train	25833	3675	1829	3168

Table 3.2: Statistics of Arabic ACE05 Train, Dev, and Test Splits.

Tokenization Tokenization described in §3.3 may result in data loss which we quantify in Table 3.2. The table shows statistics on ACE05 train, dev, and test splits. The train, dev, and test sets are identical for all approaches. This table plays allows understanding the impact of preprocessing steps on data. This table compares the number of entities, relations, triggers, and roles across different stages:

- **gold**: Represents the original data before any preprocessing.
- **segm**: Represents the data after document segmentation. The difference between textbfgold and textbfsegm rows indicates data loss due to segmentation.
- **concat**, **tok_wp**, **tok_morph**: These rows represent the data after applying the respective tokenization approaches. The difference between **segm** and these rows shows the additional data loss introduced by each tokenization method.

Dataset Statistics In Table 3.1, we present statistics done on raw ACE05 files, where NW denotes newswires, BN denotes broadcast news and WL denotes weblogs. Note that the difference between role numbers here and gold role numbers of Table 3.2 is explainable by the fact that we don't handle time roles; arguments that refer to time. We made this choice following (Luan et al., 2019) and Zhang et al. (2019). Thus we also consider that `time` and `value` event arguments are not technically named entities.

We provide the following detailed statistics about the ACE05 dataset:

- Table 3.3: This table shows the distribution of entities across different entity types (e.g., `Person`, `Organization`, `Location`).
- Table 3.4: This table presents the distribution of relations between entities (e.g., `PER-SOC` for social relations between people).
- Table 3.5: This table shows the distribution of event triggers (e.g., `Conflict`, `Transaction`).

Entity Acronym	Facility FAC	Geopolitical GPE	Location LOC	Organization ORG	Person PER	Vehicle VEH	Weapon WEA
Count	1427	7165	1215	4885	17150	418	481

Table 3.3: Statistics of Arabic ACE05 Entity Types.

Relation	ART	GEN-AFF	ORG-AFF	PART-WHLE	PER-SOC	PHYS
Count	338	1142	1379	903	643	314

Table 3.4: Statistics of Arabic ACE05 Relation Types.

Trigger	Business	Conflict	Contact	Justice	Life	Movement	Personnel	Transaction
Count	24	550	274	379	398	435	152	58

Table 3.5: Statistics of Arabic ACE05 Trigger Types.

Role	Count	Role	Count
Adjudicator	91	Origin	112
Agent	282	Organization	17
Artifact	378	Person	302
Attacker	303	Place	351
Beneficiary	22	Plaintiff	12
Buyer	6	Prosecutor	22
Defendant	135	Recipient	17
Destination	275	Seller	1
Entity	584	Target	310
Giver	36	Vehicle	50
Instrument	266	Victim	364

Table 3.6: Statistics of Arabic ACE05 Role Types.

- Table 3.6: This table details the distribution of event argument roles (e.g., `Agent`, `Patient`, `Location`).

- Table 3.7: This table highlights the top 10 most frequent entities in the Arabic ACE05 data. The total number of gold entities being 32420, we can easily see that the pronominal entities which are in most cases subwords, are numerous. Hence the need for tokenization to manage them. Note that 21.88% of entities are one-character tokens and 10.18% are two-character tokens.

Rank	Entity	Occurrences	Rank	Entity	Occurrences
1	ت (*t*)	2420	6	وا (*wA*)	459
2	ه (*h*)	1823	7	نا (*nA*)	374
3	ي (*y*)	1690	8	الرئيس (*Alr}ys; the president*)	307
4	ها (*hA*)	933	9	ن (*n*)	282
5	هم (*hm*)	560	10	أ (>)	279

Table 3.7: Top-10 Most Frequent Entities in Arabic ACE05.

3.4.2 Hyperparameters

Hyperparameter Tuning To optimize the model's performance, we use a combination of predefined hyperparameter settings and a grid search approach. We trained our model for 80 epochs with a batch size of 6. We used the BertAdam optimizer with a learning rate of 5e-5 and weight decay of 1e-5 for the BERT parameters and a learning rate of 1e-3 and weight decay of 1e-3 for other parameters.

Pretrained Language Model The primary pretrained language model we used for all experiments was *bert-large-arabertv2* by Antoun et al. (2020) except for **tok_wp** experiments, where we used the *bert-large-arabertv02* tokenizer. However, an important consideration is the potential mismatch between the tokenization schemes **tok_morph** and **tok_wp** and the vocabulary of the chosen BERT model. Currently, there isn't a pretrained BERT model specifically designed to work with these tokenization approaches.

This mismatch can lead to the model encountering tokens that are not present in its vocabulary, potentially impacting performance. Despite using *bert-large-arabertv2*, the **tok_morph** approach still resulted in a significant reduction in unknown tokens compared to the **concat** approach.

To fully address this limitation and leverage the benefits of **tok_morph** or similar tokenization schemes, a BERT language model should be trained on the output generated by the morphological analyzer. This would create a model with a vocabulary that aligns better with the tokenized data, potentially leading to improved performance.

Computational Resources We conducted our experiments on an Ubuntu machine equipped with an Nvidia GeForce RTX 2080 GPU with 8GB of RAM. We estimate the computational cost to be approximately 6 GPU hours for each experiment run of Table 3.8.

3.4.3 Evaluation

We use the standard metrics: precision, recall, and F1 score to evaluate the performance of our model on each task independently. Additionally, we calculate a macro-averaged F1 score

F_g to provide a combined measure across all tasks, where each task's weight is proportional to the number of instances N_t it contains. The formula for F_g is as follows:

$$F_g = \frac{1}{\sum_{t \in \mathcal{T}} N_t} \sum_{t \in \mathcal{T}} N_t F_1^t$$

We consider an entity (resp. trigger) correct if its span boundaries (start and end positions) and label exactly match those of a gold entity (resp. trigger). However, for subword entities (§3.3), slight mismatches within a word are tolerated as long as the overall order is preserved. For example, consider the word in Figure 3.1 tokenized using the **tok_wp** approach. If the model predicts "مراسل" *(mrAsl)* as an entity of type `Person` and "تنا" *(tnA)* as an entity of type `Organization`, this prediction would be considered correct. This evaluation approach applies to all tokenization methods.

Similar to entity evaluation, we consider a relation prediction correct if the participating entities and the relation label match the gold standard values. Likewise, we consider an event role prediction correct if its span and label align with the gold standard annotation.

While stricter evaluation is also possible, we opted for a more relaxed approach to emphasize a fair comparison between the tokenization and concatenation methods. Both approaches inherently introduce data loss due to their preprocessing steps, and our evaluation strategy accounts for this by penalizing models for the data they lose.

3.5 Results

Table 3.8 presents the results using type labels (7 entities, 6 relations, 8 triggers, and 22 roles), and Tableand 3.9 presents the results using subtype labels (44 entities, 18 relations, 32 triggers, and 22 roles) for each tokenization scheme. The scores represent the average of three model runs, with results reported for the model achieving the best average F1 score on the dev set.

Existing work on Arabic NER for ACE05 did not address nominal and pronominal entities (Benajiba et al., 2008a), avoiding the tokenization challenges. In contrast, our approach handles all grammatical categories of entity mentions.

tok_morph results The **tok_morph** approach achieves the highest F1 score on all four tasks. It also achieves the best overall F_g score. We suppose that the morphological information incorporated by the tokenizer plays a crucial role in this superior performance, particularly for relation and event recognition tasks.

concat results The **concat** approach gets the lowest F_g score. We can notice that its performance on triggers using type labels is quite close to that of **tok_morph**, but its performance on entities is poor compared to **tok_wp** and **tok_morph** approaches. We explain this by

Task	**concat**	**tok_wp**	**tok_morph**
Entity	P: 83.66 ± 0.05 R: 82.26 ± 0.11 F: 82.96 ± 0.03	P: 84.42 ± 0.32 R: 84.05 ± 0.12 F: 84.23 ± 0.22	**P: 85.04 ± 0.25** **R: 85.07 ± 0.2** **F: 85.05 ± 0.12**
Relation	P: 59.88 ± 1.29 R: 56.88 ± 0.62 F: 58.34 ± 0.94	P: 57.92 ± 1.38 R: 53.0 ± 3.02 F: 55.29 ± 1.67	**P: 62.3 ± 0.42** **R: 63.5 ± 0.61** **F: 62.9 ± 0.51**
Trigger	P: 67.56 ± 2.38 R: 58.58 ± 0.73 F: 62.74 ± 1.45	**P: 69.49 ± 0.36** R: 57.68 ± 1.89 F: 63.02 ± 1.1	P: 66.32 ± 0.51 **R: 61.11 ± 1.62** **F: 63.59 ± 0.81**
Role	P: 55.8 ± 1.09 R: 43.75 ± 0.85 F: 49.04 ± 0.95	P: 52.75 ± 0.46 R: 40.15 ± 0.81 F: 45.59 ± 0.35	**P: 57.38 ± 1.5** **R: 47.25 ± 0.94** **F: 51.82 ± 0.98**
F_g	76.31	76.66	**78.65**

Table 3.8: Results on Arabic ACE05 Data Using Type Labels.

Task	**concat**	**tok_wp**	**tok_morph**
Entity	P: 81.86 ± 0.18 R: 80.54 ± 0.32 F: 81.19 ± 0.25	P: 81.74 ± 0.22 R: 80.85 ± 0.13 F: 81.3 ± 0.18	**P: 83.05 ± 0.44** **R: 83.0 ± 0.45** **F: 83.02 ± 0.44**
Relation	P: 58.61 ± 1.56 R: 55.33 ± 1.33 F: 56.92 ± 1.41	P: 56.62 ± 0.48 R: 51.25 ± 1.0 F: 53.8 ± 0.77	**P: 60.7 ± 0.44** **R: 57.5 ± 0.5** **F: 59.05 ± 0.06**
Trigger	P: 64.93 ± 2.34 R: 55.88 ± 1.44 F: 60.06 ± 1.76	**P: 66.97 ± 0.68** **R: 56.61 ± 0.25** **F: 61.36 ± 0.14**	P: 64.32 ± 1.38 R: 54.41 ± 1.96 F: 58.96 ± 1.73
Role	P: 53.06 ± 1.07 R: 42.05 ± 1.39 F: 46.9 ± 1.03	P: 50.46 ± 2.45 R: 38.35 ± 0.57 F: 43.56 ± 1.28	**P: 55.48 ± 2.2** **R: 42.61 ± 1.14** **F: 48.2 ± 1.55**
F_g	74.50	74.03	**76.16**

Table 3.9: Results on Arabic ACE05 Data Using Subtype Labels.

the increase in the number of labels to classify in this approach; 24 entity type labels (resp. 127 entity subtype labels), such as `PER-VEH`, `ORG-VEH`, `VEH-VEH` (resp. `PER:Group-VEH:Air`, `PER:Individual-VEH:Air`), instead of 7 entity type labels (resp. 44 entity subtype labels), such as `PER`, `LOC`, `VEH`... (resp. `PER:Group`, `PER:Individual`, `VEH:Air`...) for the other

Entity	Precision	Recall	F1 Score
FAC	P: 0.86 ± 0.00	R: 0.77 ± 0.01	F: 0.82 ± 0.00
GPE	P: 0.86 ± 0.00	R: 0.84 ± 0.00	F: 0.85 ± 0.00
LOC	P: 0.79 ± 0.01	R: 0.58 ± 0.02	F: 0.66 ± 0.02
ORG	P: 0.73 ± 0.01	R: 0.79 ± 0.00	F: 0.76 ± 0.00
PER	P: 0.88 ± 0.00	R: 0.91 ± 0.00	F: 0.90 ± 0.00
VEH	P: 0.68 ± 0.00	R: 0.92 ± 0.03	F: 0.78 ± 0.01
WEA	P: 0.93 ± 0.04	R: 0.72 ± 0.03	F: 0.81 ± 0.03

Table 3.10: Entity Results by Type with **tok_morph** Tokenization.

approaches.

Relations (resp. roles) F1 score is degraded by 4.56 (resp. 2.78) points compared to that of **tok_morph** even if the relation labels number is the same for these two approaches. We explain this by the fact that when the classification and identification of entities become more complex, the part of the loss specific to entities becomes difficult to minimize, which forces the model to prioritize this task over the others, thus degrading relation and role performance.

tok_wp results Entity and relation performance of **tok_wp** is close to that of **tok_morph** and better than that of **concat**. However, this approach gets the lowest F1 score for relation and role tasks. This is partly due to a larger number of discarded entities in this approach than in the other approaches. More discarded entities leads to more discarded relations, and since we penalize each model with respect to discarded instances, this explains the discrepancy in performance.

Type labels experiments details We present in this subsection score details of the experiments of Table 3.8. Tables 3.10, 3.11, 3.12, and 3.13 shows entity, relation, trigger, and role scores by type labels.

We do not report scores details of the subtype label experiments (Table 3.9) because they are too numerous, and in general, the behavior and the performance of the subtype labels experiments follow that of the type label experiments.

We notice that among the entity types, PER has the best F1 score. Likewise, among the relation types, ORG-AFF has the best F1 score. PER and ORG-AFF represent respectively 52.87% and 29.22% of the total number of entities and relations.

Imbalanced Data Problem We notice furthermore that Business events have an F1 score of 0; they represent only 0.5% (of the total number of events), which is a limited amount of data to train the model to recognize this class. The same behavior (with an F1 score of 0) is observed for role types Beneficiary, Buyer, Organization, Prosecutor, Recipient, and Seller as they represent respectively 0.14%, 0.41%, 0.53%, 0.41%, and 0.02% of the total number

Relation	Precision	Recall	F1 Score
ART	P: 0.59 ± 0.02	R: 0.57 ± 0.04	F: 0.58 ± 0.02
GEN-AFF	P: 0.61 ± 0.02	R: 0.64 ± 0.03	F: 0.62 ± 0.02
ORG-AFF	P: 0.69 ± 0.01	R: 0.77 ± 0.01	F: 0.73 ± 0.01
PART-WHOLE	P: 0.53 ± 0.01	R: 0.6 ± 0.01	F: 0.56 ± 0.01
PER-SOC	P: 0.66 ± 0.04	R: 0.6 ± 0.01	F: 0.63 ± 0.02
PHYS	P: 0.64 ± 0.02	R: 0.21 ± 0.06	F: 0.31 ± 0.07

Table 3.11: Relation Results by Type with **tok_morph** Tokenization.

Trigger	Precision	Recall	F1 Score
Business	P: 0.00 ± 0.00	R: 0.00 ± 0.00	F: 0.00 ± 0.00
Conflict	P: 0.60 ± 0.02	R: 0.74 ± 0.03	F: 0.67 ± 0.01
Contact	P: 0.34 ± 0.02	R: 0.45 ± 0.02	F: 0.39 ± 0.02
Justice	P: 0.77 ± 0.03	R: 0.51 ± 0.02	F: 0.62 ± 0.02
Life	P: 0.80 ± 0.01	R: 0.87 ± 0.01	F: 0.84 ± 0.0
Movement	P: 0.72 ± 0.04	R: 0.29 ± 0.05	F: 0.42 ± 0.06
Personnel	P: 0.76 ± 0.04	R: 0.46 ± 0.03	F: 0.57 ± 0.03
Transaction	P: 0.74 ± 0.02	R: 0.68 ± 0.02	F: 0.71 ± 0.02

Table 3.12: Trigger Results by Type with **tok_morph** Tokenization.

of roles. For example, the Recipient role is always incorrectly predicted by the model as the Beneficiary role, since these two roles are very close semantically in the context of a Transaction event.

Comparison to other languages Table 3.5 show state-of-the-art F1 scores of joint IE with ACE05 dataset for different languages. English, Chinese, and Spanish experiments were borrowed from Lin et al. (2020), who trained their model with type labels for entity, relation, and roles, and with subtype labels for triggers. We thus give scores of Arabic following this pattern. Moreover, the presented scores are those of **tok_morph** experiments.

Overall results Unless using **concat** tokenization procedure, our model assigns one label to each input token, which establishes an upper bound on its performance since multi-label tokens are out of its reach. For example, **tok_wp** experiments could at most reach a recall of 97.31 for entities, 90.75 for relations, and 93.46 for roles; i.e., at most an F1 score of 98.63 for entities, 95.15 for relations, and 96.71 for roles.

Importantly, the performance of our three systems of Table 3.8 is comparable to other languages (Lin et al., 2020) (details in Table 3.5).

Since there was no baseline addressing the entirety of ACE05 entities, nor a system for

Role	**Precision**	**Recall**	**F1 Score**
Adjudicator	P: 0.62 ± 0.13	R: 0.27 ± 0.03	F: 0.37 ± 0.03
Agent	P: 0.49 ± 0.04	R: 0.40 ± 0.03	F: 0.44 ± 0.04
Artifact	P: 0.72 ± 0.04	R: 0.52 ± 0.04	F: 0.60 ± 0.04
Attacker	P: 0.51 ± 0.02	R: 0.60 ± 0.02	F: 0.55 ± 0.02
Beneficiary	P: 0.00 ± 0.00	R: 0.00 ± 0.00	F: 0.00 ± 0.00
Buyer	P: 0.00 ± 0.00	R: 0.00 ± 0.00	F: 0.00 ± 0.00
Defendant	P: 0.52 ± 0.11	R: 0.14 ± 0.04	F: 0.22 ± 0.06
Destination	P: 0.56 ± 0.05	R: 0.61 ± 0.04	F: 0.58 ± 0.05
Entity	P: 0.41 ± 0.02	R: 0.41 ± 0.01	F: 0.41 ± 0.00
Giver	P: 0.50 ± 0.14	R: 0.27 ± 0.09	F: 0.35 ± 0.11
Instrument	P: 0.78 ± 0.02	R: 0.63 ± 0.05	F: 0.69 ± 0.04
Origin	P: 0.65 ± 0.04	R: 0.31 ± 0.00	F: 0.42 ± 0.00
Organization	P: 0.00 ± 0.00	R: 0.00 ± 0.00	F: 0.00 ± 0.00
Person	P: 0.67 ± 0.03	R: 0.50 ± 0.06	F: 0.57 ± 0.04
Place	P: 0.50 ± 0.03	R: 0.49 ± 0.02	F: 0.49 ± 0.03
Plaintiff	P: 0.33 ± 0.47	R: 0.07 ± 0.09	F: 0.11 ± 0.16
Prosecutor	P: 0.00 ± 0.00	R: 0.00 ± 0.00	F: 0.00 ± 0.00
Recipient	P: 0.00 ± 0.00	R: 0.00 ± 0.00	F: 0.00 ± 0.00
Seller	P: 0.00 ± 0.00	R: 0.00 ± 0.00	F: 0.00 ± 0.00
Target	P: 0.55 ± 0.04	R: 0.46 ± 0.07	F: 0.50 ± 0.05
Vehicle	P: 1.00 ± 0.00	R: 1.00 ± 0.00	F: 1.00 ± 0.00
Victim	P: 0.62 ± 0.01	R: 0.74 ± 0.09	F: 0.67 ± 0.04

Table 3.13: Role Results by Type with **tok_morph** Tokenization.

Language	**Entity**	**Relation**	**Trigger**	**Role**
English	89.6	58.6	72.8	54.8
Chinese	88.5	62.4	65.6	52.0
Spanish	81.3	48.1	56.8	40.3
Arabic (Ours)	85.05	62.9	58.96	51.82

Table 3.14: State-of-the-Art F1 Scores of Joint IE for Different Languages

RE and ED, we propose **tok_morph** as a baseline.

3.6 Error Analysis And Discussion

Error analysis is important to understand the model's weaknesses and to attempt to fix them in future work. Thus, we examined a sample of 32 sentences where we found 110 remaining errors from experiments using the **tok_morph** tokenization scheme and using the type labels.

Entity Errors A significant portion of errors (23%) involve pronominal entities. These include two types of issues:

- **Missed Pronouns**: The model fails to predict entities present in the gold data. Consider the word “صادرتها” (*SAdrthA; confiscated it*). In this case, the pronoun “ت” (*t*) is annotated in gold data as a PER entity, which the model fails to predict. These errors are most likely due to the absence of labeling for a considerable number of pronominal entities in the gold data. An illustrative example is found in the word “المسلحين” (*AlmslHyn; armed*), where the model incorrectly predicts the pronoun “ين” (*yn*) as a PER entity, despite its absence in the gold data. However, it’s worth noting that this pronoun was annotated 167 times in words like “المتقاعدين” (*AlmtqAEdyn; retirees*), “الآخرين” (*AlAxryn; the others*), and “الراغبين” (*AlrAgbyn; willing to*).
- **Misclassified Pronouns**: These are errors of correctly identified entities being misclassified.

It’s important to note that pronominal entities comprise a substantial 31% of the total gold entities, highlighting their importance and the potential impact of limited labeling on model performance.

Relation Errors Two primary categories contribute to relation errors (14% of total errors):

- **Missed Relations with Multiple Entities**: These errors occur when entities participate in multiple relations within the same sentence. For instance, consider the gold annotations of the sentence “وزير العدل المصري” (*wzyr AlEdl AlmSry; Egyptian Minister of Justice*). In this example, the word “وزير” (*wzyr; Minister*) is involved in two distinct relations: one of type ORG-AFF with the word “العدل” (*AlEdl; Justice*), and another of type GEN-AFF, with the word “المصري” (*AlmSry; Egyptian*). However, the model only predicts the first ORG-AFF relation between the initial two words, overlooking the additional relation.
- **Misclassified Relation Types**: This category (at least 6% of errors) involves correctly identified entity pairs but with an incorrect relation type assigned. This often arises due to semantic ambiguity between certain relation types, particularly those with overlapping contexts. For instance, distinguishing between PART-WHOLE and ORG-AFF relations can be challenging due to their overlapping semantic contexts.

Figure 3.2 presents some examples of remaining relation errors for visualization purposes.

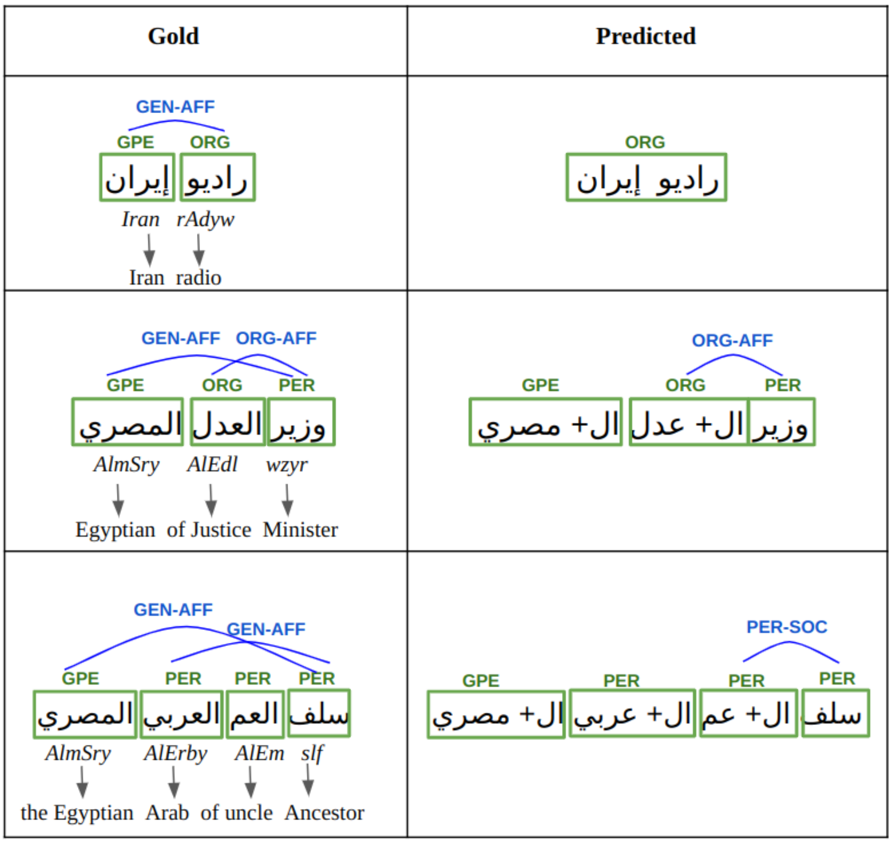


Figure 3.2: Examples of Remaining Relation Errors.

Events Errors Approximately 23.5% of the errors identified in the analysis pertains to events, specifically triggers and roles. Among the 35 remaining event errors, a significant portion (67%) can be attributed to **annotation omissions**, highlighting the need for a thorough examination of the model's performance in event detection and classification.

As an example, in the sentence "اتصل به شقيقه" (*AtSl bh $qyqyh; his brothers called him*), the model predicts the verb "اتصل" (*AtSl; called*) as a trigger of type `Contact`. This trigger is not annotated in the gold data but the model's prediction seems correct because an event of type `Contact` is defined in the annotation guide by: explicit phone or written communication between two or more parties. In the annotation guide the verb called in the sentence "John called Jane last night" is given as an example of a trigger of type `Contact`.

Figure 3.3 presents a recurring example of a long sentence containing several omitted roles. In this sentence, we distinguish three errors: (1) the word "المتهمين" (*Almthmyn; The accused*) is predicted as an `Agent` argument by the model, which is intuitively correct as an `Agent` is defined in the annotation guide by: the attacking agent or the one that enacts the harm. This word is incorrectly annotated in the gold sentence as an argument of type `Victim`. (2) The word "رفاق" (*rfAq; companions*) is predicted as an argument of type `Agent` which is intuitively correct. This word is not annotated in the gold sentence as an argument. (3) The word "الصائغ" (*AlSAg; the jeweler*) is predicted as arguments of type `Victim` which is intuitively correct as a `Victim` is defined in the annotation guide by: the person who died.

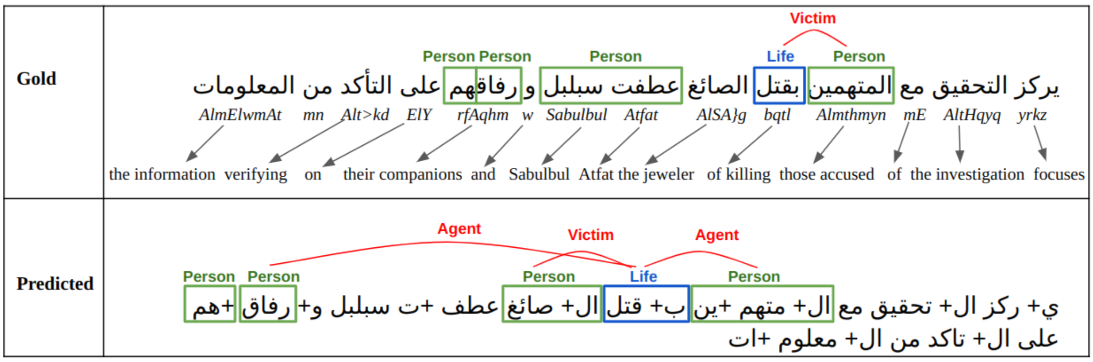

Gold
Victim
Person Person
Person
Life
Person
يركز التحقيق مع المتهمين بقتل الصائغ عطفت سبلبل و رفاقهم على التأكد من المعلومات
AlmElwmAt mn Alt>kd ElY rfAqhm w Sabulbul Atfat AlSA}g bqtl Almthmyn mE AltHqyq yrkz
the information verifying on their companions and Sabulbul Atfat the jeweler of killing those accused of the investigation focuses
Predicted
Agent
Victim
Agent
Person Person
Person
Life
Person
ي+ ركز ال+ تحقيق مع ال+ متهم +ين ب+ قتل ال+ صائغ عطف +ت سبلبل و+ رفاق +هم
على ال+ تاكد من ال+ معلو +ات

Figure 3.3: Examples of Remaining Event Errors Due to Annotation Omissions

This word is not annotated in the gold sentence.

3.7 Related Work

Entity Extraction Most Arabic IE work focuses on NER. We cite Naji (2012), who used artificial neural networks for NER. Oudah and Shaalan (2012) tested a hybrid approach, including both rule-based and machine learning approaches. Benajiba et al. (2008b) proposed an SVM-based model with a combination of language-dependent and language-dependent features, showing the relevance of morphological features for rich languages like Arabic. Benajiba et al. (2010) built a system augmented by deeper lexical, syntactic, and morphological features that were extracted from noisy data obtained via projection from an Arabic-English parallel corpus. Helwe et al. (2020) proposed a semi-supervised learning approach to train a BERT-based NER model using labeled and semi-labeled datasets. The works that deal with NER using ACE05, ACE04, or ACE03 either preprocess the data differently from ours, which results in a very different number of entities than ours or use different entity types than the one we used. For example, Benajiba et al. (2008b) evaluate their model separately for each data type of ACE05 (NW, BN, WL). In addition, they remove all annotations that they consider not oriented to the entity detection and recognition tasks, such as the nominal and pronominal entities, and only keep the named ones, which leads them to a total number of entities in the training and test corpora of 10218. This makes their performance incomparable to ours because we evaluate the model with almost 32000 entities for all our proposed approaches. Other work use Benajiba et al. (2010, 2009, 2008a) the same preprocessing of Benajiba et al. (2008b). Oudah and Shaalan (2012) tested their model performance on Date, Time, Price, Measurement, and Percent entities of ACE05, while we test our model on the principal entity types (`PER, LOC, ORG, FAC, VEH`...).

Relation Extraction Arabic RE works include Mohamed et al. (2015), who proposed a distant supervised learning model with specific features that characterize Arabic relations. Sarhan et al. (2016) presented a semi-supervised pattern-based bootstrapping technique for relation extraction using stemming and semantic expansion. Taghizadeh et al. (2018) used a combination of kernel functions and the universal dependency parsing for supervised relation extraction. We can't compare our work to these as relation extremities (entities) are already recognized in a NER preprocessing, while we extract all information jointly.

Event Extraction Very little work has been done on event extraction; AL-Smadi and Qawasmeh (2016) proposed a knowledge-based approach for event extraction on Arabic tweets. And Alsaedi and Burnap (2015) proposed a classification/ clustering-based framework to detect real-world events from Twitter. Ahmad et al. (2021) developed a Graph Attention Transformer Encoder to generate structured contextual representations for cross-lingual relation and event extraction working on ACE05. Yet, they haven't addressed the problem of the mismatch between the tokenization and the annotations; problematic entities were simply discarded.

3.8 Conclusion and Discussion

Our work presents the first joint IE model for Arabic, achieving performance comparable to models for other languages. We also delve into the challenge of subword entities, prevalent in morphologically rich languages like Arabic, and propose two approaches to address them. Our findings demonstrate the importance of incorporating morphological information for accurate subword entity recognition. Our key findings are the following:

- **Morphological Tokenizer Superiority**: The morphological tokenization approach consistently outperforms the other approaches across all tasks. We hypothesize that the morphological information captured by this tokenizer empowers the model to better grasp complex relations and event structures. This highlights the crucial role of morphology in Arabic IE tasks.
- **Impact of Annotation Omissions**: Error analysis revealed a significant portion of errors, particularly in event detection tasks, can be attributed to annotation inconsistencies within the gold standard data. While the model makes some predictions that differ from the gold annotations, these predictions may, in some cases, be intuitively correct based on the provided definitions of event types and roles. This finding underscores the need for further investigation into model performance beyond relying solely on gold standard accuracy metrics.

This work represents a significant step forward in Arabic NLP and aims to establish a strong foundation for further advancements in Arabic IE tasks and within the Arabic NLP community. However, this work acknowledges some limitations that present opportunities for future exploration:

- **Random Entity Selection**: As described in Section 3.3, after the tokenization process, if a subword still holds multiple entities, our model currently selects one randomly and discards the others. This process leads to considerable data loss. Future work should investigate more sophisticated methods for selecting the most relevant entity or even explore incorporating all potential entities. Character-based tokenization might be a promising direction to explore for this purpose.

- **Tokenization-Vocabulary Mismatch**: Another problem related to the tokenization process is that of the mismatch between the vocabulary generated by the tokenizers and the BERT vocabulary used for token encoding (cf. Section 3.4.2). A potential solution involves training a custom BERT model specifically on the output of the chosen tokenizer.

- **Limited Inter-Task Communication**: Although our model uses multitask learning during training through the loss funtion and the shared token encodings, it does not explicitly account for the interdependencies between tasks within the information graph, as greedy search is used as a decoding strategy. The current greedy search decoding strategy only selects the highest-scoring element for each instance. Exploring alternative decoding strategies that allow for considering a wider range of possibilities during search could potentially improve performance.

Chapter 4

Cross-Dialectal Named Entity Recognition in Arabic

4.1 Introduction

In this chapter, we address the crucial challenge of Named Entity Recognition (NER) in Arabic dialects. While our previous work (Chapter 3) has demonstrably achieved significant progress in information extraction for Arabic, it focused on Modern Standard Arabic MSA due to data availability. This focus limits its applicability to the vast and growing amount of Arabic text written in dialects.

Linguistic Diversity in Arabic Dialects and Challenges Arabic is renowned for its rich linguistic diversity, with over 20 distinct dialects and approximately 100 regional variants spoken across the Arab world. These dialects are widely used in everyday communication, particularly in digital spaces. The rise of social media platforms and online communication in Arabic dialects necessitates tools capable of understanding these rich linguistic variations. This emphasizes the urgent need for NLP models that can effectively handle this linguistic diversity.

However, this diversity poses unique challenges in the context ofNER, including:

- **Linguistic Variation**: Arabic dialects exhibit significant linguistic variation in terms of phonology, morphology, syntax, and lexicon. This variation can pose challenges to developing unified global modeling NER approach, as the same entity may be represented differently across different dialects. Consider the word "Car". In MSA, the word for "*Car*" is "سيارة" (*syArp*), while in Moroccan Darija, the commonly used term is "طنو بيل" (*Tnwbyl*). In Tunisian Darija, the commonly used term is "كرهبة" (*krhbp*), and in Saudian dialect, they use the term is "مركبة" (*krhbp*).
- **Scarcity of annotated data**: Annotated data for Arabic dialects is scarce compared to

MSA. The limited availability of labeled datasets constrains the training and evaluation ofNER models, as machine learning algorithms typically require large amounts of annotated data for good performance. We discuss the few existing datasets in Section 4.7.

Leveraging Cross-Lingual Transfer Learning Our research is driven by the goal of bridging the linguistic gap between MSA and Arabic dialects, specifically in the context of entity extraction. To do so, we leverage a successful technique from related areas of NLP: cross-lingual transfer learning.

In cross-lingual transfer learning, knowledge is transferred from a well-resourced language with abundant training data, such as MSA in our case, to a low-resource language with limited annotated data, such as Arabic dialects. This approach has demonstrated strong performance in cross-lingual entity extraction, particularly when using English as the source language and targeting languages like Spanish, German, and MSA (Chen et al., 2021b; Wu et al., 2020a; Jain et al., 2019a). The ACE05 dataset (Walker and Consortium, 2005) has also proven to be a valuable resource for cross-lingual information extraction (Ahmad et al., 2021; Subburathinam et al., 2019; Nguyen et al., 2021b) from English as a source language to target languages such as Spanish, Chinese, and MSA, as it provides annotations for entity, relation, and event extraction for these four languages following the same guidelines. The cross-lingual entity extraction approach allows us to exploit the capabilities of models trained on MSA data and adapt them to dialectal text, even with limited dialectal annotations.

Contributions This work makes the following key contributions:

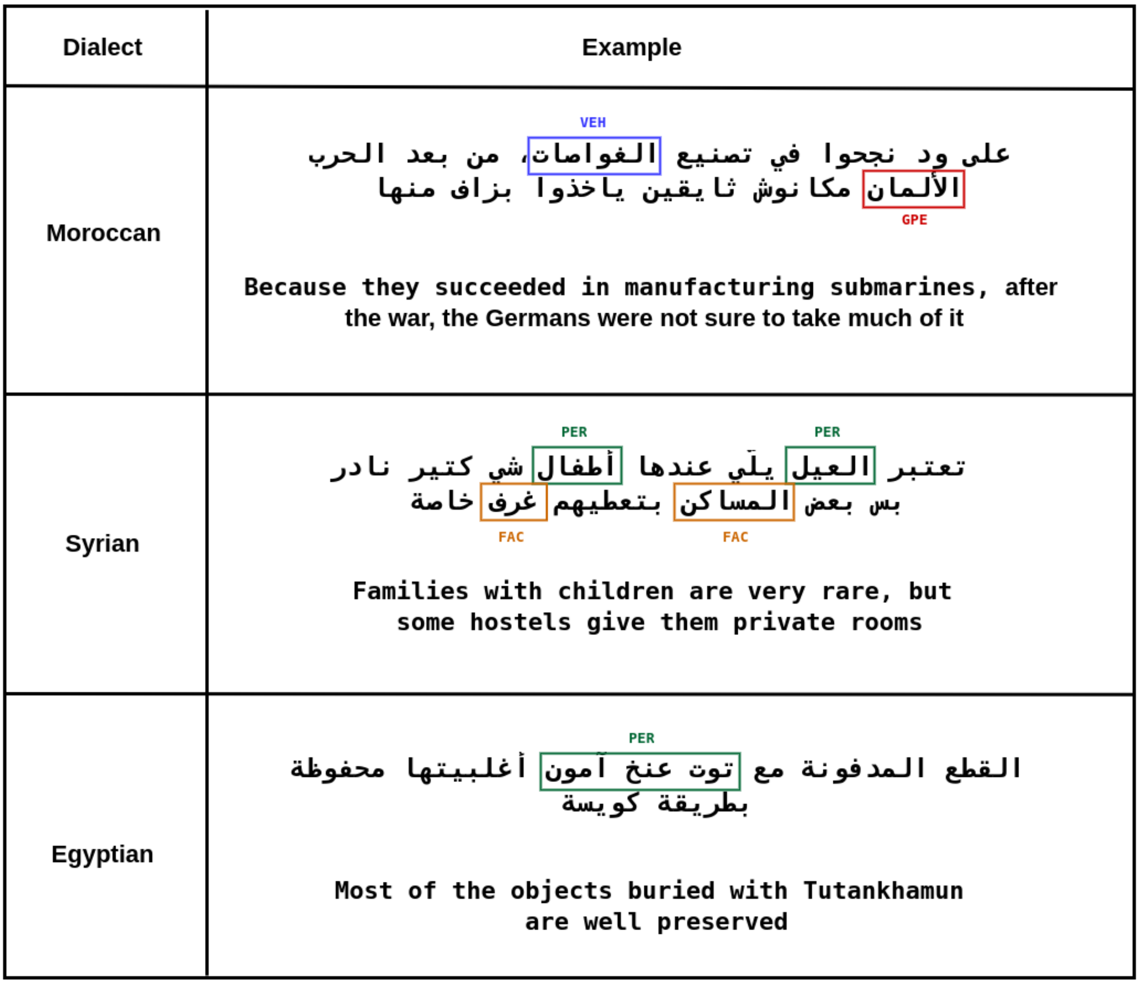
- **Creation of dialects Dataset**: We introduce a NER dataset manually annotated for three dialects: Moroccan, Egyptian, and Syrian. This dataset is used for evaluation purposes;
- **Extensive experimentation**: We train an efficient span-basedNER model on already-available MSA data and analyze its transferability to other dialects. We also explore the inverse setting where we train on dialects and evaluate their transferability to MSA.

4.2 Dataset and Annotation

In this section, we introduce the datasets used in our work: Modern Standard Arabic dataset and Arabic Dialects datasets (Moroccan, Egyptian, Syrian). We detail their construction processes and the annotation guidelines employed.

4.2.1 Modern Standard Arabic Dataset

Data Source and Selection Our MSA dataset leverages the Arabic Corpus ACE 2005 (Walker and Consortium, 2005). This corpus offers a rich collection of text data from di-


VEH
GPE
PER
PER
FAC
FAC
PER

Dialect	Example
Moroccan	على ود نجحوا في تصنيع الغواصات، من بعد الحرب الألمان مكانوش ثايقين ياخذوا بزاف منها Because they succeeded in manufacturing submarines, after the war, the Germans were not sure to take much of it
Syrian	تعتبر العيل يلّي عندها أطفال شي كتير نادر بس بعض المساكن بتعطيهم غرف خاصة Families with children are very rare, but some hostels give them private rooms
Egyptian	القطع المدفونة مع توت عنخ آمون أغلبيتها محفوظة بطريقة كويسة Most of the objects buried with Tutankhamun are well preserved

Figure 4.1: Example of Annotations from our Dialect Dataset.

verse sources like news wires, broadcast news, and weblogs. It includes annotations for seven entity types (e.g., Person, Location), three entity mention types (e.g., Name, Nominal Construction), and coreference information.

Our Modern Standard Arabic dataset is sourced from the Arabic Corpus ACE05 (Walker and Consortium, 2005). The ACE corpus comprises a rich collection of text data from diverse sources, including newswires, broadcast news, and weblogs. As described in Chapter 2, this corpus includes annotations for seven distinct entity types, namely persons (PER), organizations (ORG), geopolitical entities (GPE), locations (LOC), facilities (FAC), vehicles (VEH), and weapons (WEA). In addition to entity types, it annotates three entity mention types: names (NAM), nominal constructions (NOM), and pronouns (PRO). The corpus offers annotations for both flat and nested entities, further including coreference information.

We opted to focus on a subset of the ACE 2005 corpus, specifically targeting sentences relevant to NER. Here is a breakdown of our selection choices:

- **Focus on NAM and NOM entities**: We opted to concentrate exclusively on the recognition of named entities and nominal constructions while excluding pronouns. ACE 2005 is notable for its detailed annotation, including pronouns, which is uncommon in the typical named entity recognition task that primarily deals with nominal entities and names. Pronoun usage exhibits considerable variation, displaying nuanced distinctions not only between dialects but even within distinct regions of the same dialect. Consequently, accurately annotating pronouns across dialects presents practical challenges and potential ambiguity, due to their strong contextual reliance and the absence of comprehensive dialect-specific guidelines. The inclusion of pronouns is therefore left to future work. For clarity, named entities include examples such as "جون" (*John*) and "رام االله" (*rAm AAllh; Ramallah*), while nominal entities include examples like "المحامي" (*AlmHAmy; The lawyer*) and "ميناء" (*mynA'; Port*). Pronominal entities, which we chose to exclude, include terms such as "هم" (*hm; they*), "بعض" (*bED; some*), and "كثيرون" (*kvyrwn; many*).

- **Focus on flat entities**: We opted to concentrate exclusively on flat entities, omitting nested entities and coreference resolution. This choice simplifies the task significantly by reducing complexity in both annotation and modeling. Nesting and coreference, while valuable areas of study, introduce intricate challenges, especially in dialectal Arabic, where linguistic variations are prevalent. Focusing on flat entities streamlines our research process, making it more scalable for testing across dialects.

Dataset Construction We also extracted an additional 350 MSA sentences to train an MSA model and evaluate it on the 500 sentences for reference. More details can be found in the results section (5.5)

Based on these methodological decisions, we constructed our MSA dataset by randomly selecting 500 sentences from the ACE05 corpus. Tables 4.1 and 4.2 (first columns) provide detailed statistics about these sentences. This dataset serves two purposes:

1. Training a model to analyze its transferability to dialects.
2. Evaluating models trained on dialectal datasets.

For reference purposes, we created an additional 350 MSA sentence dataset for training a dedicated MSA model. This model's performance will be evaluated on the original 500-sentence dataset. The results section (Section 5.5) will provide more details on this additional dataset.

4.2.2 Annotation Guidelines for Dialects

We introduce concise yet comprehensive annotation guidelines that were used in the annotation of our dialectal datasets. These guidelines closely follow the ACE guidelines that were used for the MSA dataset. The detailed reference is provided by the Linguistic Data Consortium (LDC) guidelines[1].

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace/annotation-tasks-and-specifications

1. **Person (PER)**: This entity type is used for individual human beings. It includes:
   - Names and surnames of individuals. *Example*: “ميت رومني” (*myt rwmny; Mitt Romney*)
   - Group of people. *Example*: “العائلة” (*AlEA}lp; The family*).
   - Saints and other religious figures. *Example*: “ٱللّه” (*{lla ‘h; God*).
2. **Organization (ORG)**: This entity type is used for corporations, agencies, and other groups of people defined by an organization structure. It includes:
   - Commercial organizations. *Example*: “ميكروسوفت” (*mykrwswft; Microsoft*)
   - Government organizations. *Example*: “البحرية الملكية“ (*AlbHryp Almlkyp; Royal Navy*).
   - Educational organizations. *Example*: “جامعة ستانفورد” (*jAmEp stAnfwrd; Stanford University*).
   - Political parties. *Example*: “الحزب الليبرالي” (*AlHzb AllybrAly; Liberal Party*).
   - Media. *Example*: “وكالة انسا” (*wkAlp AnsA; ANSA agency*).
3. **Location (LOC)**: This entity type is used for geographical entities such as mountains, rivers, seas, and regions that aren’t politically defined. *Example*: “شمال نيو مكسيكو” ($mAl nyw mksykw; Northern New Mexico).
4. **Geographical/Social/Political Entity (GPE)**: This entity type is used for geographical regions that have a political distinction. This includes countries, states, provinces, and cities. *Example*: “أمريكا” (*mrykA; America*).
5. VEH (Vehicle): This entity type is used for entities that are primarily designed for transporting goods or people from one place to another. *Example*: “عربة” (*Erbp; vehicle*).
6. Weapon (WEA): This entity type is used for devices used with intent to inflict damage or harm.
   - Exploding. *Example*: “قنابل” (*qnAbl; Bombs*).
   - Chemical. *Example*: “الغاز” (*AlgAz; Gas*).
   - Underspecified. *Example*: “سلاح” (*slAH; Weapon*).
7. FAC (Facility): This entity type is used for buildings or structures. It includes buildings, houses, factories, stadiums, office buildings, gymnasiums, prisons, museums, space stations, barns, parking garages and airplane hangars, streets, highways, airports, ports, train stations, bridges, and tunnels. *Example*: “المطار” (*AlmTAr; The airport*).

We adhere to these guidelines by annotating the smallest constituent of flat entities. For example, consider the entity “بطل الولايات المتحدة” (*United States champion*). In this case, we annotate الولايات المتحدة (*bTl AlwlAyAt AlmtHdp; United States*) as GPE and “بطل” (*bTl; champion*) as PER. If our task involved nested entities, we would have provided additional annotations for the entire nested entity “بطل الولايات المتحدة” as PER.

4.2.3 Annotation Process of the Dialect Datasets

Our dataset for Arabic Dialects is sourced from the xP3x corpus Muennighoff et al. (2022). The xP3x corpus comprises a vast collection of prompts and datasets across 277 languages, covering 16 distinct NLP tasks. We specifically used the section containing sentence pairs and their translations in three Arabic dialects: Moroccan, Egyptian, and Syrian. To ensure the accuracy and reliability of our annotations, we followed these steps:

- **Data Selection**: We randomly selected 500 sentences from the xP3x corpus for each dialect and tokenized them using whitespace.
- **Annotator Training**: Our annotation process was supervised by a single proficient annotator, with native fluency in the Moroccan dialect and possessing a strong grasp of Egyptian and Syrian dialects. The annotator received comprehensive training on the annotation guidelines, including real-world examples of dialectal variations and potential disambiguation challenges.
- **Annotation Tool**: We used a web-based annotation tool called **Label Studio**[2] widely used forNER tasks. This tool provides an intuitive interface for annotators to efficiently highlight relevant text spans and assign the corresponding entity type.

Given the limited dataset size, employing a single annotator was advantageous for maintaining consistency, coherence, and manageable workloads, thereby reducing inter-annotator discrepancies and ensuring uniform annotation styles.

4.3 Task Definition and Modeling

In this section, we provide a detailed overview of theNER task and the model architecture employed for this task.

Dataset statistics After the annotation process, we only retained sentences containing entities for our experiments.

Table 4.1 summarizes key statistics for each dialectal dataset, including the number of sentences, tokens, and total named entities identified. As can be observed, MSA exhibits a

[2] https://labelstud.io/

Stat	**MSA**	**Mor.**	**Egy.**	**Syr.**
Sentences	500	378	353	361
Tokens	14168	6780	6533	6034
Entities	3030	970	831	956

Table 4.1: Dialect Dataset Statistics.

Ent	**MSA**	**Mor.**	**Egy.**	**Syr.**
FAC	143	83	63	71
GPE	923	249	229	331
LOC	160	191	142	89
ORG	413	112	77	109
PER	1269	278	264	307
VEH	52	45	50	41
WEA	70	12	6	8

Table 4.2: Dialect Dataset Statistics by Entity Type.

higher number of entities compared to the dialectal datasets. This can be attributed to factors such as the formal nature of MSA texts often containing more explicit references to named entities compared to informal communication channels found in dialectal data.

Table 4.2 provides a detailed breakdown of the distribution of named entity types across each dialectal dataset. The most frequent entity types include person (PER), geopolitical entities (GPE), and organizations (ORG), reflecting the inherent nature of language where these categories are commonly referenced in text.

Figure 4.1 showcases examples of annotated sentences from our dialectal dataset.

Named Entity Recognition as Sequence Labeling Following the work of Zaratiana et al. (2022a,b), we frame the task ofNER as a span classification problem. Given an input sequence: $\boldsymbol{x} = \{x_i\}_{i=1}^{L}$, our objective is to classify all potential spans within the sequence, defined as:

$$\boldsymbol{y} = \bigcup_{i=1}^{L} \bigcup_{j=i}^{L} s_{ijc} \tag{4.1}$$

Here, i, j, and c correspond to the start position, end position, and span type, respectively. The probability of a specific span classification $\boldsymbol{y}$ given the input sequence $\boldsymbol{x}$ is represented as:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp \sum_{s_{ijc} \in \boldsymbol{y}} \phi_\theta(s_{ijc}|\boldsymbol{x})}{\mathcal{Z}_\theta(\boldsymbol{x})} \tag{4.2}$$

In this equation, $\phi_\theta(.)$ is the span scoring function, and $\mathcal{Z}_\theta(\boldsymbol{x})$ is the partition function. During training, our objective is to minimize the negative log-likelihood of the gold span classifications.

Training loss During training, our assumption allows us to bypass the need to explicitly evaluate the partition function $Z_\theta(\boldsymbol{x})$ to compute the loss. The loss for a single sample $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}$ is simply the sum of loss for all spans in the input:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{c_{ij} \in \boldsymbol{y}} \log p(c_{ij}|\boldsymbol{x}) \tag{4.3}$$

where,

$$p(c_{ij}|\boldsymbol{x}) = \frac{\exp \phi_\theta(c_{ij}|\boldsymbol{x})}{\sum_{c' \in \mathcal{C}} \exp \phi_\theta(c'_{ij}|\boldsymbol{x})} \tag{4.4}$$

This loss is minimized over the training set using a stochastic gradient descent algorithm.

Decoding During inference, our aim is to determine:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \sum_{s_{ijc} \in \boldsymbol{y}} \phi_\theta(s_{ijc}|\boldsymbol{x}) \tag{4.5}$$

In other words, we seek to identify the span labeling configuration that achieves the highest score. For unconstrained span classification, a straightforward approach is to assign the label with the highest score to each individual span, as follows:

$$s_{ijc^*} = \arg\max_{c} \phi_\theta(s_{ijc}|\boldsymbol{x}) \tag{4.6}$$

Nonetheless, this decoding approach is not optimal since it may result in structural constraint violations. In our context of flat entities, overlapping entity spans are strictly prohibited. A more efficient solution, as presented in (Zaratiana et al., 2022a,b), employs a two-stage decoding process. Initially, spans predicted as non-entities are filtered out, followed by the application of a maximum independent set algorithm to the remaining spans to determine the optimal set of entity spans.

Token and Span Representations We compute the span score $\phi_\theta(s_{ijc}|\boldsymbol{x})$ by performing a linear projection of the span representation, which is derived from a $1D$ convolution applied to token representations obtained from a transformer-based model (eg. BERT):

$$\boldsymbol{s}_{ijc} := w_c^T \mathsf{Conv1D}_k([\boldsymbol{h}_i; \boldsymbol{h}_{i+1}; \ldots; \boldsymbol{h}_j]) \tag{4.7}$$

Here, $h_i \in \mathbb{R}^D$ represents the token representation at position i, k signifies the size of the convolutional filter (corresponding to the span length), and $w_c \in \mathbb{R}^D$ denotes a learned weight matrix associated with span label c.

4.4 Experimental Setup

In this section, we detail the experimental setup, the employed hyperparameters, and the evaluation metrics for ourNER model on Arabic dialects.

4.4.1 Token Encodings

To encode our input tokens, we leverage a comprehensive set of eight pretrained language models, each exhibiting unique characteristics and training data sources. Here's a breakdown of the chosen PLMs:

- **Modern Standard Arabic (MSA) Focused Models**:
  - ***ARBERTv2***: (Abdul-Mageed et al., 2021): A large-scale pretrained masked language model for MSA with 12 attention layers, 12 heads, 768 hidden dimensions, and 163M parameters, trained on 61GB of Arabic text.
  - ***CAMeLBERT-MSA*** (Inoue et al., 2021): A collection of pretrained BERT models for MSA, trained on a diverse dataset of 107GB, totaling 12.6 billion tokens.
- **Dialectal Arabic Focused Models**:
  - ***MARBERTv2*** (Abdul-Mageed et al., 2021): A large-scale pretrained masked language model for both DA and MSA, trained on 1B Arabic tweets (128GB text, 15.6B tokens), using the same architecture as ARBERT (BERT-base) without next sentence prediction.
  - ***CAMeLBERT-DA*** (Inoue et al., 2021): A collection of pretrained BERT models for Arabic dialects, trained on a diverse dataset of 54GB, totaling 5.8 billion tokens.
- **Mixed MSA and Dialect Models**:
  - ***AraBERTv2*** (Antoun et al., 2020): The dataset consists of 77GB Arabic text from diverse sources. It uses the same architecture as BERT-Base.
  - ***CAMeLBERT-Mix*** (Inoue et al., 2021): A collection of pretrained BERT models for Arabic, including MSA, DA, and CA, trained on a diverse dataset of 167GB, totaling 17.3 billion tokens.
- **Multilingual Models**:

- ***mBERT*** (Devlin et al., 2019): The multilingual version of BERT pretrained on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective.
- ***mDeBERTav3***: A multilingual version of DeBERTa (He et al., 2020) trained with CC100 multilingual data.

By employing this diverse set of pretrained language models, we aim to explore their effectiveness in handling named entity recognition tasks across the chosen Arabic dialects (Moroccan, Syrian, Egyptian).

4.4.2 Hyperparameters

Hyperparameter Tuning To optimize the model's performance, we use a combination of predefined hyperparameter settings and a grid search approach. Here's an overview of the key hyperparameters and their chosen values:

- **Batch Sizes**: We use a training batch size of 12 and a validation batch size of 32.
- **Learning rates**: We use a learning rate of `2e-5` for the pretrained parameters and a learning rate of `3e-3` for the other parameters. We train all our models up to convergence. For testing, we use the last model, given the limited availability of validation data in our dataset.
- **Maximum Span Length**: To manage the complexity of the task, we impose a constraint on the maximum span length, setting it to a maximum width of $K = 10$. This constraint significantly reduces the number of segments from L^2 to LK.

Training Hardware and Software Libraries

- The training process is conducted on a server equipped with V100 GPUs.
- We leverage the AllenNLP library for efficient data preprocessing tasks.
- The pretrained transformer models are conveniently loaded from the HuggingFace Transformers library.

4.4.3 Evaluation Metrics

We adopt the standardNER evaluation methodology, calculating precision (P), recall (R), and F1 score (F), based on the exact match between predicted and actual entities.

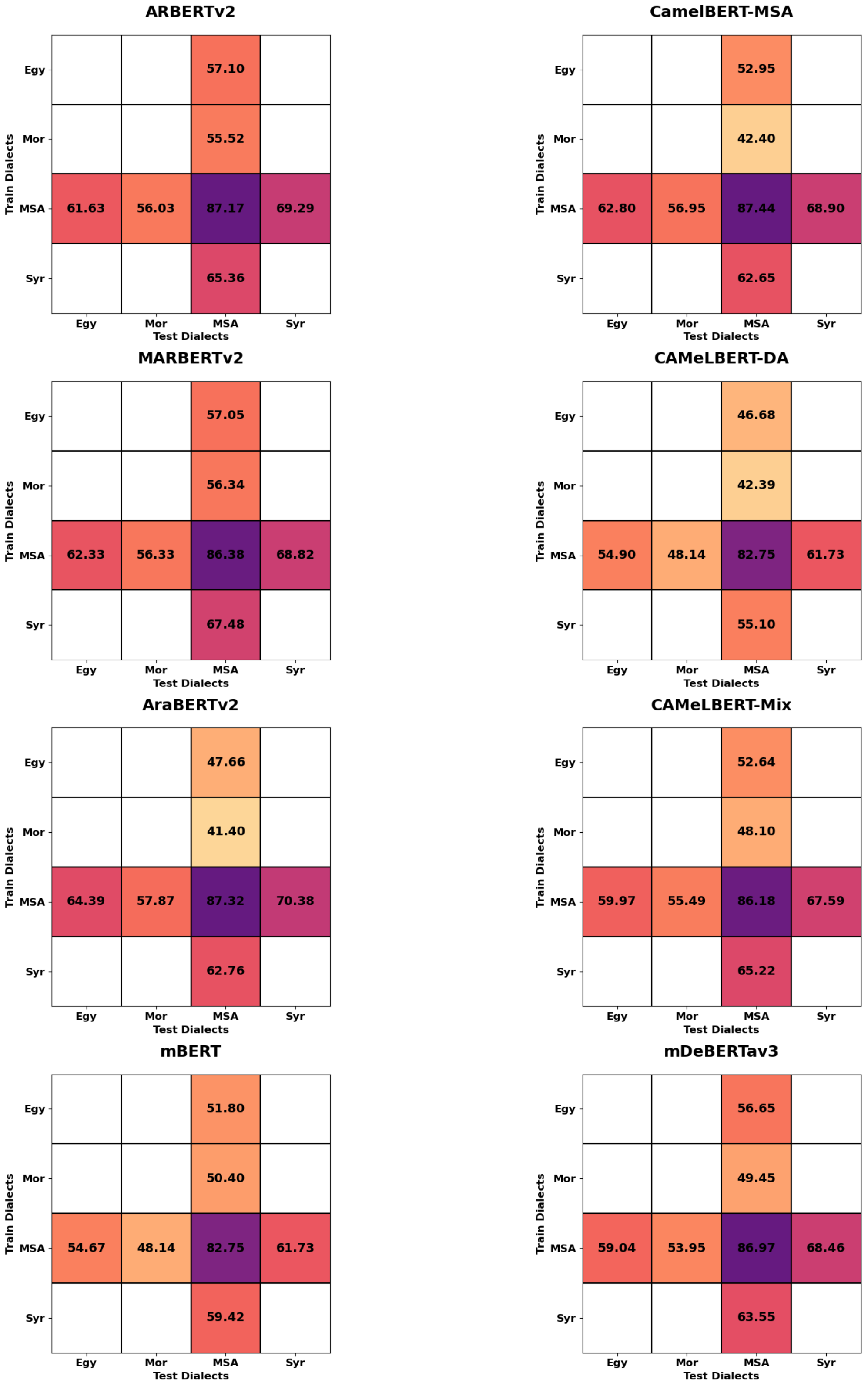


Figure 4.2: Performance Comparison of Models Across Various Training and Testing Settings (F1 Score).

4.5 Results

We conducted two primary experiments for theNER task: firstly, training on Modern Standard Arabic, and evaluating on dialects, and secondly, reversing this configuration, training on individual dialects, and assessing on MSA. For both scenarios, we used the complete dataset outlined in Table 4.1.

In addition, we conducted MSA-to-MSA experiments, where we evaluated our model on the MSA dataset specified in Table 4.1, while the training set consisted of a random selection of 350 sentences drawn from the original Arabic ACE dataset, using the same preprocessing steps detailed in Section 4.2.1. The main results of our experiments are shown in Figure 4.2.

4.5.1 Main Results

Train on MSA and Test on MSA Here, both training and testing data consist of MSA text. The performance metrics reveal that MSA-to-MSA settings consistently yield the highest accuracy across all tested configurations, a result that aligns with expectations given that Modern Standard Arabic often serves as the benchmark for Arabic language tasks. Interestingly, most backbone models such as *ARBERTv2*, *mDeBERTav3*, *CAMeLBERT-MSA*, *CAMeLBERT-Mix*, *AraBERTv2*, and *MARBERTv2* demonstrate comparable performance with F1 scores around 87 and 86, suggesting that their architecture and training data are well-suited for MSA-centric tasks.

Two models, however, diverge from this trend. *CAMeLBERT-DA* exhibits a 4% drop in performance compared to the other language models, which can be attributed to its focus on dialectal data during training. This specialization likely limits its ability to generalize effectively to MSA. Similarly, *mBERT* performs less well, with a 4% drop in performance compared to the other language models. As a multilingual model, *mBERT* may suffer from language interference or tokenization issues, given its training on a diverse corpus where Arabic is not the dominant language.

Train on MSA and Test on Dialects When training models on the MSA dataset, the observed performance metrics indicate a hierarchical trend among the tested Arabic dialects. The best performances are systematically obtained with the Syrian dialect, followed by the Egyptian dialect, and finally the Moroccan dialect. This gradient could be indicative of the linguistic similarities and differences between MSA and these dialects. The Syrian dialect may share more syntactic and semantic features with MSA, allowing models trained on MSA to generalize more easily to Syrian. On the other hand, the Moroccan dialect appears to be the most divergent from MSA among the tested dialects, resulting in the lowest performance scores. This could be due to unique lexical, grammatical, or even phonological features that are not adequately captured when a model is trained solely on MSA data.

Train on Dialects and Test on MSA Similar to the MSA to dialects scenario, the best test performance on MSA is obtained when models are trained on the Syrian dialect, followed

Model	**Avg. F1**
ARBERTv2	68.53
MARBERTv2	68.46
CamelBERT-MSA	69.02
CAMeLBERT-DA	61.88
AraBERTv2	70.00
CAMeLBERT-Mix	67.31
mBERT	61.82
mDeBERTav3	67.10

Table 4.3: Average F1 score by Language Model Trained on MSA.

by the Egyptian dialect and finally the Moroccan dialect. This pattern aligns well with the earlier observation that models trained on MSA perform best on the Syrian dialect, thereby suggesting a mutual linguistic affinity between Syrian and MSA. Models trained on Egyptian also perform relatively well, reinforcing the notion of shared linguistic features between Egyptian and MSA. Conversely, the Moroccan dialect, which was identified as the most challenging for models trained on MSA, also proves to be the least effective training data for models tested on MSA. This consistent underperformance across both scenarios could point to a greater linguistic divergence between Moroccan and MSA, which may involve lexical, syntactic, or phonological differences not easily bridged by the models in question.

We acknowledge that we conducted additional experiments training on one dialect and testing on others. However, we are not presenting those scores in this chapter. The reason for this is that the source dataset used to create our dialectal datasets contains some sentences that are direct translations between dialects. This overlap between datasets could lead to misleading performance metrics when evaluating transfer between dialects.

4.5.2 Optimal PLM Model for each setting

Optimal Language Model for MSA Training As shown in Table 4.3, when training with the MSA dataset, AraBERTv2 emerges as the top-performing language model, with an average score of 70.00 across various Arabic dialects. The strength of this model can be attributed to its well-balanced training regimen, which combines both MSA and dialectal data, resulting in a harmonious blend of specialization and generalization.

Models explicitly trained on MSA, namely ARBERTv2 and CAMeLBERT-MSA, closely follow in terms of performance, underscoring the effectiveness of MSA-focused training. The lowest performance is achieved by mBERT and CAMeLBERT, possibly due to language interference issues.

Overall, our data suggests that a balanced training approach, as exemplified by AraBERTv2, offers the most effective strategy for tasks involving MSA and its various dialects.

Dialect	Best Model	Avg. F1
Egyptian	AraBERTv2	64.39
Moroccan	AraBERTv2	57.87
Syrian	AraBERTv2	70.38

Table 4.4: Best-Performing Language Model for Test Dialect.

Dialect	Best Model	Avg. F1
Egyptian	ArBERTv2	57.10
Moroccan	MARBERTv2	56.34
Syrian	MARBERTv2	67.48

Table 4.5: Best-Performing Language Model for Train Dialect.

Entity	Egyptian	Moroccan	MSA	Syrian
FAC	45.33	46.43	79.02	59.26
PER	71.25	63.74	89.54	77.56
ORG	50.24	54.30	78.13	52.02
GPE	78.57	68.03	90.93	80.12
LOC	45.02	34.58	77.91	43.90
WEA	50.00	57.14	94.20	50.00
VEH	66.02	59.26	83.17	75.27
Avg	64.39	57.87	87.32	70.38

Table 4.6: F1 Scores by Entity Type for Dialects Trained with *AraBERTv2* on MSA.

Optimal Language Models for Each Dialect Table 4.4 shows the best-performing PLM in the setting of training on MSA and testing on a specific dialect, and Table 4.5 shows the best-performing PLM in the setting of training on a specific dialect and testing on MSA.

However, Table 4.5 reveals a different picture when considering models trained on a specific dialect for NER on MSA. Here, *AraBERTv2* only performs best for tasks involving Egyptian Arabic. For Moroccan and Syrian dialects, *MARBERTv2* shows superior performance in transferring knowledge to identify named entities in MSA.

4.6 Error Analysis

In this section, we conduct an error analysis of the output of our system.

Perfomance by Entity Type We first focus on the performance breakdown by entity type. We present the detailed F1 scores achieved for each entity type in Table 4.6. The table shows the results of experiments training on MSA data and testing on each specific dialect (Egyptian, Moroccan, and Syrian) using *AraBERTv2* as a token encoder. We observe the following:

- **High Performance for PER and GPE**: Entities of type PER and GPE achieve F1 scores above 70% for most cases and across all dialects. Reasons the model effectively identifies these entity types even with dialectal variations include their inherent salience in language, as they refer to specific and well-defined concepts. Names of people and places tend to be less susceptible to grammatical or morphological changes across dialects compared to other words. Moreover, these entity types are well-represented in the training data, compared to other entity types. This abundance of training examples allows the model to effectively learn the patterns and features associated with these entities.

- **Low Performance for ORG and VEH**: Entities of type ORG and VEH achieve F1 scores below 50% for most cases and across all dialects. Reasons the model struggles to recognize these entities include greater dialectal variation, as organizations and vehicle names might exhibit more significant variations in vocabulary or naming conventions across dialects compared to PER and GPE entities.

Entity Types Confusion Matrix To further investigate these results, we focus on Moroccan results as they are the lowest. We present in Figure 4.3 the confusion matrix in terms of percentages for entity types, with rows presenting true entity types and columns presenting the predicted ones. This matrix reveals the type of errors where the model incorrectly assigns labels to entities while correctly identifying their span offsets. These errors, where the model predicts an entity but assigns the wrong type, represent 64.07% of the total number of entities predicted by the model.

We observe a high performance for most entity types, such as PER, GPE, VEH and WEA. However, it's important to consider the limited sample size for WEA (12 entities in total). With such a small number of examples, this result might not be statistically significant.

We also observe that the model struggles the most with entities of type location (LOC), often misclassified as geopolitical entities (GPE) and facilities. Moreover, even with decent performance for entities of type GPE and ORG and LOC, there an ambiguity associated with them that leads to 31% of misclassified entity errors. These errors often concern country or city names, such as “الولايات المتحدة” (*bTl AlwlAyAt AlmtHdp; United States*) which, depending on the context, may belong to any of these categories.

4.7 Related Work

For languages like modern standard Arabic, a significant body of research exists onNER. However, due to the inherent dialectal richness of Arabic, there is growing interest in devel-

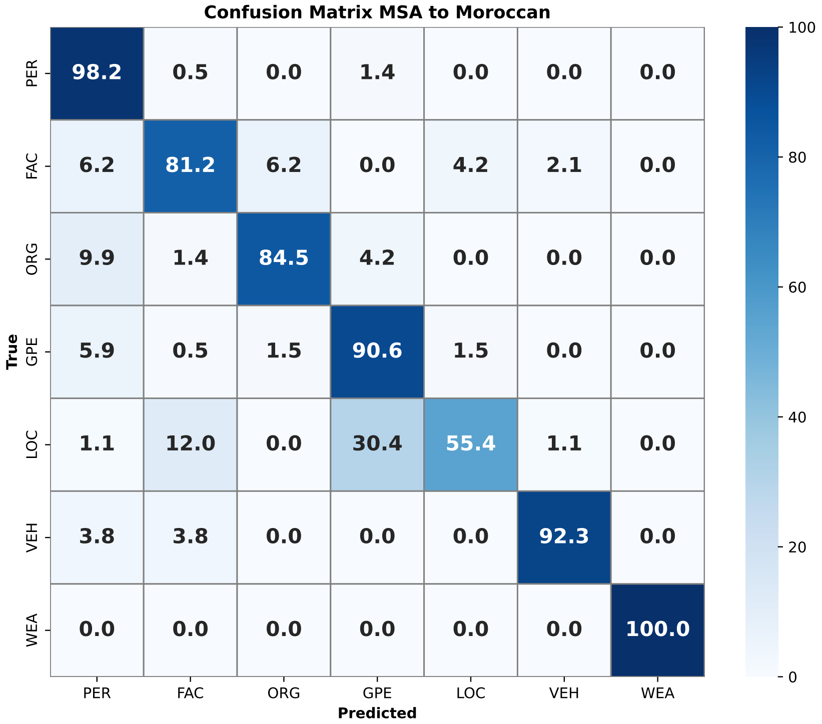


Figure 4.3: Entity Type Confusion Matrix, Training on MSA and Testing on Moroccan, Using *AraBERTv2.*.

opingNER models that can handle the variations present in spoken Arabic dialects. These dialects differ from MSA in vocabulary, morphology, and syntax, posing unique challenges for NER tasks.

Transfer learning has emerged as a promising approach for addressing these challenges. By leveraging knowledge gained from pretrained language models on MSA data, researchers can developNER models that can perform well on dialect-specific datasets with limited annotated data. This section will review existing research onNER for Arabic dialects and the application of transfer learning techniques in this domain.

4.7.1 Datasets and Named Entity Recognition for MSA

NER for MSA The development of Named Entity Recognition techniques in Modern Standard Arabic has been a central focus within the Arabic NLP community. Initially, rule-based NER systems like those described in Shaalan and Raza (2008); Abdallah et al. (2012) relied on manually crafted grammatical rules and gazetteers. While effective, these systems demanded extensive maintenance and lacked scalability. Subsequently, machine learning-based NER methods, as demonstrated by Benajiba and Rosso (2007); Al-Qurishi and Souissi (2021), treated NER as a classification task, leveraging large annotated datasets. This era also witnessed the fusion of rule-based and machine learning-based approaches through hybrid systems Oudah and Shaalan (2012); Meselhi et al. (2014), followed by the adoption of deep learning techniques, which allowed for the automatic extraction of intricate features. Deep learning, characterized by neural networks processing word and character embeddings, marked a departure from manual feature engineering, resulting in significantly improved accuracy and a more streamlined approach to ArabicNER. In recent years, pretrained language models such as BERT (Devlin et al., 2019) have opened up a new era in Arabic NER. Arabic-specific PLMs, such as AraBERT (Antoun et al., 2020) and AraELECTRA (Antoun et al., 2021), have been meticulously developed and fine-tuned for NER tasks, offering the advantage of context-rich information. This evolution has given rise to a multitude of high-performance systems (Helwe et al., 2020; El Khbir et al., 2022).

NER Datasets for MSA Additionally, extensive annotation efforts have led to the creation of high-quality MSA NER datasets. ACE 2005 (Walker and Consortium, 2005) comprises a diverse text collection with annotations for seven entity types (PER, ORG, GPE, LOC, FAC, VEH, WEA), three mention types (NAM, NOM, PRO), and coreference information. ANERcorp (Benajiba et al., 2007) comprises articles from diverse sources. It includes traditional entity types (ORG, LOC, PER) and introduces a MISC (miscellaneous) type. AQMAR (Mohit et al., 2012) comprises hand-annotated text extracted from Arabic Wikipedia articles. It includes 28 articles categorized by domain, each tagged with named entities and custom entity classes. Wojood (Jarrar et al., 2022) comprises text sourced from different domains and manually annotated with 21 entity types, including both flat and nested entities.

4.7.2 Datasets and Named Entity Recognition for Arabic dialects

Few works addressedNER for Arabic dialects. Zirikly and Diab (2014) introduced an annotated dataset and a named entity recognition system tailored to the Egyptian dialect. However, their evaluation focused solely on two entity types: PER and LOC. In a subsequent work, Zirikly and Diab (2015) presented a gazetteer-freeNER system tailored to the Egyptian dialect, evaluated on three entity types: PER, LOC, and ORG. Additionally, Moussa and Mourhir (2023) introduced a manually annotatedNER dataset for the Moroccan dialect, which comprises 4 entity types: PER, LOC, ORG and MISC.

4.7.3 Cross-Lingual NER

Cross-lingual named entity recognition (CLNER) has been largely explored for languages with limited resources. A core challenge in cross-lingual NER is the scarcity of labeled data for many languages. To address this, researchers have adopted knowledge transfer techniques, leveraging well-annotated data in a high-resource source language, such as English, to train models that can operate on a low-resource target language, such as Spanish or Arabic.

Successful knowledge transfer relies on resources that bridge the gap between languages. Commonly used resources include:

- **Parallel Corpora** (Fu et al., 2011; Ehrmann et al., 2011): Aligned text collections in both source and target languages provide a gold standard for understanding how languages express similar concepts. However, creating these resources can be expensive and time-consuming.

- **Pseudo-Parallel Corpora** (Qian et al., 2014): Machine translation can be used to generate corpora where source and target language sentences are aligned, offering a more scalable alternative to parallel corpora. However, the quality of the translation system directly impacts the usefulness of this resource.

- **Multilingual Word Embeddings and Language Models**: These pretrained models capture semantic similarities between words across languages, allowing models to learn language-independent representations. Examples include Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), and PLMs such as *mBERT* (Devlin et al., 2019).

There are two main approaches to knowledge transfer in CLNER:

- **Data Transfer**: These methods project labels from the annotated source data onto unlabeled target data. This often involves parallel corpora or machine translation to achieve alignment between source and target languages. Related work includes Jain et al. (2019b) who leverage machine translation and bilingual dictionaries to perform NER data transfer via annotation projection from English to Armenian, German, Spanish, Hindi, and Chinese.

  While data transfer methods can leverage target-language specific features, the quality of the alignment process heavily influences their performance.

- **Direct Transfer**: In contrast, direct transfer methods train models solely on source language data. These methods rely on techniques like shared representations or pretrained multilingual language models to learn language-independent features that can be applied to the target language. Related work includes that of Bari et al. (2020) who transfer NER knowledge from English to Arabic, German, Spanish, and Dutch using bilingual embeddings and direct transfer via adversarial learning. Wu et al. (2020b) also leverage direct NER transfer from English to German, Dutch, Spanish, French, and Chinese, using multilingual PLMs and meta-learning.

  While this approach avoids the need for target-language labeled data, it may struggle to capture target-specific information and can be less effective for languages with significant typological differences.

In addition, researchers are exploring ways to combine the strengths of data and direct transfer approaches. Hybrid methods might leverage unlabeled target data alongside source training data to incorporate some target-language information. Additionally, knowledge distillation and multi-source training are being investigated to further enhance cross-lingual learning.

Our Work In contrast to prior work that often relies on parallel corpora or machine translation for knowledge transfer between distant languages, our approach leverages an existing annotated MSA dataset for NER. We then create annotated datasets for three Arabic dialects (Moroccan, Egyptian, and Syrian) following the same guidelines and labels from ACE05. Our approach can be considered as a **data transfer approach**, specifically leveraging projection from a well-annotated source language (MSA) to low-resource target dialects (Moroccan, Egyptian, and Syrian). However, our work focuses on a single language family, Arabic, where dialects share significant morphological and syntactic similarities. This reduces the challenges associated with cross-lingual alignment typically encountered in data transfer between distant languages. We also explore PLMs trained with different data sources, some are MSA-focused, others dialectal Arabic-focused, and also multilingual models.

Our work contributes to the exploration of dialect-specific NER within CLNER, demonstrating the potential of data transfer within a single language family, particularly for languages with significant typological similarities.

4.8 Conclusion and Discussion

Our work investigated the effectiveness of transfer learning for named entity recognition in Arabic dialects, specifically focusing on transferring knowledge from modern standard Arabic to Egyptian, Moroccan, and Syrian dialects. We employed a range of pretrained language models and annotated a dataset encompassing these dialects to evaluate their performance. Our key findings are the following:

- **Syrian Dialect Affinity with MSA**: Models trained on MSA and tested on Syrian data consistently achieved the highest F1 scores across different PLMs. This suggests a strong linguistic similarity between Syrian Arabic and MSA, potentially due to historical and cultural factors.
- **Egyptian Dialect Performance**: Models trained on MSA and tested on Egyptian data also demonstrated promising results, indicating a closer connection to MSA compared to Moroccan.
- **Moroccan Dialect Challenges**: The Moroccan dialect consistently presented the most difficulty for all models. This is likely due to its significant linguistic divergence from MSA, including unique vocabulary, morphology, and syntax.
- **Model Performance Variation**: PLMs like *CAMeLBERT-DA*, *mBERT*, and *mdBERTav3* consistently underperformed compared to other models.

This work represents a significant step forward in Arabic NLP and aims to establish a strong foundation for further advancements in Arabic IE tasks and within the Arabic NLP community. However, this work acknowledges some limitations that present opportunities for future exploration:

- **No Code-Switching**: The annotated dataset did not include code-switching, a common phenomenon in real-world Arabic communication where speakers switch between languages, such as French or English, and dialects within a sentence. Developing NER models robust to code-switching scenarios will be crucial for real-world applications.

- **Limited Dialect Scope**: While this work explored three dialects, including a wider range of Arabic dialects in future studies would provide a more comprehensive understanding of how NER performance varies across the Arabic dialect spectrum. This can shed light on the generalizability of transfer learning approaches for ArabicNER tasks.

- **Single Annotator Bias**: The annotation of our dataset relies on a single annotator, which may be a potential source of bias. Future work should consider the involvement of multiple annotators to assess inter-annotator agreement and ensure labeling robustness.

- **Nested Entity Recognition**: Investigating nested NER tasks, where entities can be contained within other entities, would further challenge and potentially improve model performance.

By addressing these limitations and pursuing the proposed future work directions, we can contribute to the development of more robust and generalizable NER models for a wider range of Arabic dialects.

Chapter 5

Information Extraction with Differentiable Beam Search on Graph RNNs

5.1 Introduction

The output of the information extraction task is often structured as a *labeled graph*. In this graph, entities and triggers are represented by nodes, relations by edges joining two entity nodes, and roles by edges joining a trigger node and an entity node.

Despite significant advancements in information extraction techniques, some challenges persist, particularly in modeling the intricate dependencies between labels. Existing approaches in the literature have explored various strategies to address this issue. For instance, globally-normalized CRF-based scoring functions have been employed to model inter-instance and inter-label dependencies effectively (Yu et al., 2019). Another approach involves using auto-regressive frameworks, which build on previous decisions to make better predictions. This includes the work of (Luan et al., 2019) and (Wadden et al., 2019) which uses graph convolution layers to iteratively refine node representations, although they still use independent classifiers for labeling. Alternatively, other auto-regressive frameworks rely on sequence labeling models with Recurrent Neural Networks (RNNs) or on Sequence-to-Sequence models with Transformers. These models use specialized vocabularies to encode the labeled graph, helping to capture the dependencies between different elements more effectively (Paolini et al., 2021; Lu et al., 2022; Fei et al., 2022; Liu et al., 2022).

Challenges in Modeling Label Dependencies Training autoregressive sequence models typically involves maximizing the likelihood of each token in the reference (gold standard) sequence given previous reference tokens. During inference, however, the unknown previous tokens are replaced by model predictions, creating a discrepancy. The training scenario relies on accurate past information, while inference uses potentially erroneous predictions. This discrepancy is known as exposure bias.

While solutions like schedule sampling (Bengio et al., 2015) attempt to bridge the gap between training and inference by incorporating previous decoding decisions stochastically during training, they introduce discontinuities in the training objective. This is because they rely on greedy decisions at each time step, making it difficult for the model to learn effectively using gradient-based methods.

Furthermore, when beam search, a more sophisticated decoding technique, is used instead of greedy decoding, the objective function does not directly *reason* about the behavior of the decoder at inference time. As a result, beam decoding can sometimes yield reduced test performance when compared with greedy decoding (Cho et al., 2014a; Koehn and Knowles, 2017).

To address this challenge, previous work proposed a training objective that takes the search process into account. These methods use continuous approximations of both greedy and beam search, making the decoding stage differentiable and thus compatible with gradient-based learning (Goyal et al., 2017, 2018). This allows the model to be "aware" of its decoding behavior during training, leading to better performance in tasks like named entity recognition and segmentation. However, this approach has yet to be applied to the more general case of graph generation, which is what motivates our work.

Our Proposed Model In this work, we present a novel approach to information extraction that incorporates differentiable beam search. This technique offers several advantages that address the challenges of modeling label dependencies in IE:

- **Exploration of Diverse Candidates**: It allows the model to explore a wider range of possible label sequences while considering the intricate dependencies between labels.
- **Training-Inference Alignment**: By incorporating a continuous relaxation of beam search into training, we mitigate exposure bias.

Here's a breakdown of our model:

1. **Entity and Trigger Identification**: Similar to previous work (Lin et al., 2020), we first identify entities and triggers using a linear-chain CRF (Lafferty et al., 2001) with a BIO tagging scheme.
2. **Autoregressive Decoding**: Unlike Lin et al. (2020) which uses a combination of local classifiers with manually designed feature-based representation of the graph, we apply RNNs on the *linearized* graph You et al. (2018) to sequentially assign labels to identified nodes and potential edges between them.
3. **Differentiable Beam Search Decoding**: We use beam search during decoding to explore a wider range of possible label sequences. We show that the discrepancy between training and decoding is harmful. To address exposure bias, we introduce a continuous relaxation of beam search similar to Goyal et al. (2018). This allows the model to

be aware of its decoding behavior during training, leading to potentially better performance.

Contributions This work makes the following key contributions:

- We are the first to apply differentiable beam search to the more general case of information extraction involving graph generation (§5.3).
- We demonstrate the effectiveness of our decoding-aware model through experiments on various datasets (ACE05 and CoNLL04) and languages (English, Arabic, Chinese) (§5.5).
- We perform ablation studies to confirm the importance of reducing the gap between training and inference for optimal performance (§5.6.1 and §5.6.2).
- We propose a method for quantifying exposure bias, offering deeper insights into the behavior of our model (§5.6.3).

5.2 Task Definition

Information extraction (IE) involves identifying and labeling entities, relations, triggers, and their arguments in text data, mapping it to a labeled graph $G = (V, E)$. V is the set of nodes corresponding to entities and triggers, and E is the set of edges corresponding to relations between pairs of entities or between a trigger and one of its arguments. Each graph element (a node or an edge) is assigned a label from a set of possible types as depicted in Figure 5.1.

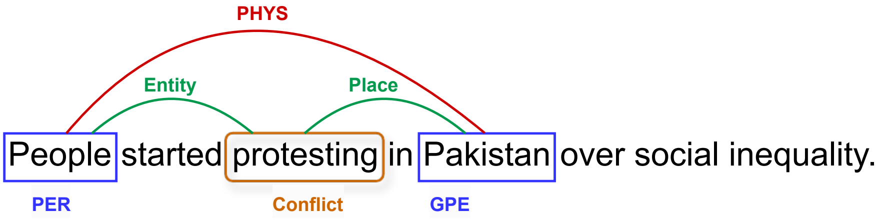

PHYS
Entity
Place
People started protesting in Pakistan over social inequality.
PER
Conflict
GPE

Figure 5.1: Example of an Information Extraction graph.

5.3 Model

Our model consists of two systems trained simultaneously in a multitask setup. The first system focuses on identifying nodes of the graph using a CRF for sequence labeling (§5.3.1). The second system tackles the generation of the labeled graph using an auto-regressive network. To achieve this, we first use the identified nodes from the previous step to construct a linearized graph structure (see Section 5.3.2). Subsequently, this linearized graph is labeled using an auto-regressive model (RNN) for representation. The decoding process relies on beam search (§5.3.3). Additionally, in Section 5.3.4, we elaborate on our approach of relaxing the beam search, which is crucial for our search-aware training procedure.

5.3.1 Node Identification

Text Encoding Initially, the input sequence is passed through a pretrained language model (PLM), such as BERT (Devlin et al., 2019). This process aims to generate a vector representation for each word in the sequence. Notably, if a word is fragmented into multiple word pieces during tokenization, we ensure consistency by considering its representation as the average of all its constituent word piece vectors.

Identification as Sequence Labeling Subsequently, the sequence of embeddings is passed through a feed-forward layer and then fed to a CRF Lafferty et al. (2001) layer. The role of the CRF here is to label the sequence using the BIO (Beginning, Inside, Outside) scheme, thus facilitating the identification of spans of tokens corresponding to entities or triggers. To accommodate potential overlaps between entities and triggers, we employ two distinct CRFs.

Example: Considering the sentence depicted in Figure 5.1, the entity CRF yields the sequence <`B, O, B, O, O`>, while the trigger CRF yields <`O, O, O, O, B`>.

Training and Inference In the training phase, we optimize the model parameters by minimizing the negative log-likelihood (L_{id}) of the reference BIO tag sequence. This loss function, L_{id}, constitutes a crucial part of the joint-training loss of our model. During inference, we employ the Viterbi algorithm to search for the most likely tag sequence.

5.3.2 Graph Linearization

In our graph representation, nodes are represented by $V = e_1, \ldots, e_n, t_1, \ldots, t_m$, where e_i denotes entities and t_i denotes triggers that have been previously identified in the input text. To ensure a consistent ordering, entities are organized based on their appearance in the sentence ($e_1, \ldots, e_n$), followed by triggers in a similar order ($t_1, \ldots, t_m$).

To predict the types of entities, triggers, relations, and arguments, we consider all possible pairwise relations and arguments, denoted as $E = \{(e_i, e_j) \in V^2\}_{1 \leq i < j \leq n} \cup \{(t_i, e_j) \in V^2\}_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$. These pairs are treated as an ordered sequence using lexicographic order. We construct the linearized graph sequence using the entity, relation, trigger, and argument sequences according to the following procedure: we iterate over the entity sequence, and at each step, we add the current entity and all relations between it and the previously added entities. This ensures that each relation appears after its two endpoints. Subsequently, we

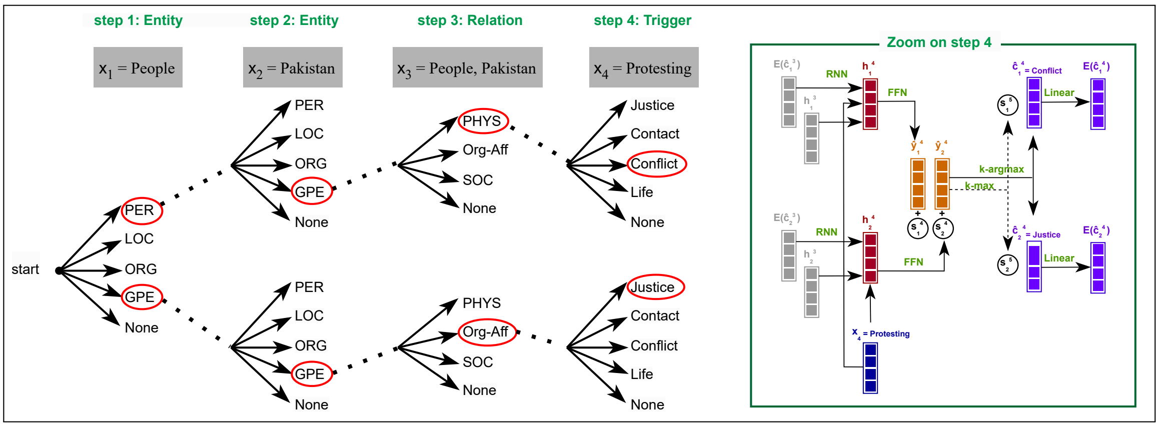


Figure 5.2: SLBS example for Figure 5.1, $K = 2$, $|\mathcal{V}_{entity}| = |\mathcal{V}_{trigger}| = 5$, and $|\mathcal{V}_{relation}| = 4$. Hidden state h_1^3 encodes the following graph path tags: *PER, GPE, PHYS* and h_2^3 encodes: *GPE, GPE, Org-Aff.*

iterate over the trigger sequence, adding at each step the visited trigger and then all possible arguments.

The resulting sequence is of length $T = n + \frac{n(n-1)}{2} + m + nm$, where n, m, $\frac{n(n-1)}{2}$, and nm respectively represent the number of entities, triggers, relations, and arguments. The sequence follows a specific ordering: e_1, e_2, (e_1, e_2), e_3, $\dots$, (e_{n-1}, e_n), t_1, (t_1, e_1), (t_1, e_2), $\dots$, (t_m, e_n).

Example: For instance, considering the linearization of the graph depicted in Figure 5.1, we obtain the following sequence: *"People"*,*"Pakistan"*, (*"People", "Pakistan"*), *"Protesting"*, (*"Protesting", "People"*), (*"Protesting", "Pakistan"*).

During training, this sequence is constructed using gold entities and gold triggers extracted from the input sentence.

5.3.3 Graph RNN with Beam Search

Encoding of Nodes and Edges The representation of a node is computed of as the average of its token representations. The representation of an edge is constructed by concatenating the representations of its two connected nodes. We denote the encoded sequence as $x \in \mathbb{R}^{d_x \times T}$, and for ease of readability, we denote x^i its i-th element in all the following.

Labeling Given the linearized graph $\mathbf{x}$, we aim to generate a label sequence $\hat{\mathbf{c}}$ of the same length, where each element x^i is assigned a label from the corresponding task vocabulary $\mathcal{V}_{task}$, where $task$ refers to one of the four IE tasks (entity, relation, trigger, and role). To account for graph elements that need to be removed, we include a dedicated `None` label in each $\mathcal{V}_{task}$.

Sequence Labeling with Beam Search (SLBS), with a beam size K, is a heuristic that approximates the most likely label sequence by keeping track of and updating K candidate sequences at each step. At each step $t = \{1, \dots, T\}$, we keep track of K couples $\{(h_i^t \in \mathbb{R}^{d_h}, s_i^t \in \mathbb{R})\}_{1 \leq i \leq K}$. The vector h_i^t can be understood as an embedding of the i-th beam element of the usual beam search algorithm, and is updated using a Recurrent Neural Network as follows:

$$h_i^{t+1} = \text{RNN}(x^{t+1}, \mathcal{E}(\hat{c}_i^t), h_i^t) \tag{5.1}$$

where $x^{t+1} \in \mathbb{R}^{d_x}$ is the current instance embedding, $\mathcal{E}(\hat{c}_i^t)$ is the embedding of $\hat{c}_i^t$ implemented as a linear projection layer, with $\hat{c}_i^t \in \{1, \dots, |\mathcal{V}_{task}|\}$ being the index of the previously selected tag, i.e. the one that best extends the i-th element of the beam. The term "best" is defined in this context using the extension scores $\tilde{s}_{i,j}^t \in \mathbb{R}$ such that, for every beam index $1 \leq k \leq K$:

$$s_k^{t+1} = \underset{\substack{1 \leq i \leq K \\ 1 \leq j \leq |\mathcal{V}_{task}|}}{\texttt{top-k-max}}(\tilde{s}_{i,j}^t) \tag{5.2}$$

$$b_k^t, \hat{c}_k^t = \underset{\substack{1 \leq i \leq K \\ 1 \leq j \leq |\mathcal{V}_{task}|}}{\texttt{top-k-argmax}}(\tilde{s}_{i,j}^t) \tag{5.3}$$

With

$$\tilde{s}_{i,j}^t = s_i^t + \hat{y}_{i,j}^t \tag{5.4}$$

The local scores $\hat{y}_{i,j}^t \in \mathbb{R}$ represent classification logits produced by feed forward networks FFN$_{task}$ when fed the hidden states h_i^t:

$$\hat{y}_{i,\cdot}^t = \text{FFN}_{task}(h_i^t) \in \mathbb{R}^{|\mathcal{V}_{task}|} \tag{5.5}$$

The local score $\hat{y}_{i,j}^t$ can be seen as the negative log-likelihood of the beam element i having j as a tag at time step t.

In equation 5.3, $b_k^t \in \{1, \dots, K\}$ serve as back-pointers because they point to the beam element whose extension produced the current state of the beam element k.

In practice, updates are made in the following order: 5.5, 5.4, 5.3, 5.2 / 5.1. Figure 5.2 illustrates an example of the first 4 steps of the SLBS procedure.

Training During training, K = 1. Hence, the model is greedily trained to minimize the total cross-entropy L_g loss at each time step between the predicted tags and the gold ones:

$$L_g = -\sum_{t=1}^{T} \sum_{j=1}^{|\mathcal{V}_{task}|} y_j^t \log(\sigma(\hat{y}_{i,j}^t)) \tag{5.6}$$

Where $y_\cdot^t \in \mathbb{R}^{|\mathcal{V}_{task}|}$ is the gold tag in its one-hot form, and σ is the softmax function.

Total Loss The model is jointly trained to minimize the nodes identification loss and the labeled graph generation loss: $L = L_{id} + L_g$.

5.3.4 Continuous Relaxation of Beam Search

The SLBS procedure is used as a decoding strategy with models that are trained greedily using cross-entropy. Hence, the distribution of hidden states reached during inference does not match that of the hidden states reached during training. In order to incorporate awareness of the decoding strategy into the training stage, we train our model using a relaxed SLBS procedure, by replacing the discontinuous `top-k-argmax` operation with the relaxed version used by Goyal et al. (2018); Maddison et al. (2017); Jang et al. (2017); Goyal et al. (2017) in the context of Seq2Seq models.

The following describes how we relax the SLBS procedure for IE, making it fully continuous and almost everywhere differentiable.

Continuous `top-k-argmax` The key ingredient is to replace the only discontinuous operation of the SLBS procedure, namely the `top-k-argmax` operation applied to extension scores, with a continuous approximation, taking advantage of the following asymptotic property: for any real-valued function f defined over the vocabulary $\mathcal{V}_{task}$, the expression $\sigma(-\frac{(f(\cdot)-m_k)^2}{\alpha})_j = \frac{e^{\frac{-(f(j)-m_k)^2}{\alpha}}}{\sum\limits_{l=1}^{|\mathcal{V}task|} e^{\frac{-(f(l)-m_k)^2}{\alpha}}}$ tends to $\delta_j(\underset{1\leq l\leq|\mathcal{V}task|}{\texttt{top-k-argmax}}(f(l)))$ as the temperature parameter α tends to zero, with δ_j being the Dirac distribution centered on the tag j, which can also be seen as the one-hot operation, and:

$$m_k = \underset{1\leq l\leq|\mathcal{V}_{task}|}{\texttt{top-k-max}}(f(l)) \tag{5.7}$$

Training with soft SLBS In the SLBS procedure, the `top-k-argmax` operation is used to make tag choices $\hat{c}_k^t$ based on the extension scores $\tilde{s}_{i,j}^t$. In the relaxed setup, a tag choice is no longer a binary decision. Therefore, using the previous asymptotic approximation, we define $p_{i,j}^k$ as the set of probability distributions over tags j (cf. lines 8 and 9 of Algorithm 1) that can be interpreted as the probability of beam element k being updated using the hidden state coming from beam element i and extended by tag j.

Such a set of probability distributions can be first used to compute a relaxed version of s_k^{t+1}, as the expected extension score over all origin beam elements i and extension tags j (cf. line 11 of Algorithm 1), and then to compute a relaxed version of the one-hot representation of the previously added tag $\hat{c}_k^t$, denoted $\hat{c}_{j,k}^t$, as the probability of j being the last tag added to the beam element k (cf. line 13 of Algorithm 1).

Loss Computation Importantly, this set of probability distributions can be used to compute the negative log-likelihood of each tag in the gold sequence, which is a problem-adapted local loss:

$$l^t = -\log P(j_*^t) = -\log(\sum_{k=1}^{K} d_k(\sum_{i=1}^{K} p_{i,j_*^t}^k)) \tag{5.8}$$

where j_*^t denotes the index of the gold tag at time step t, $\sum_{i=1}^{K} p_{i,j_t^*}^k$ represents the (marginalized) probability of j_*^t being the predicted tag given a beam k, that also can be interpreted as a posterior over the set of beams $1, \ldots, K$, and d_k being a prior over the set of beam elements. Overall, we associate the labeled graph generation with the following global loss:

$$L_{cl} = \sum_{t=1}^{T} l^t \tag{5.9}$$

Unfortunately, empirical observations show numerical instability in the computation of l^t. To address this issue, one possible approach is to tightly bound it with a term that can be stabilized using techniques such as the log-sum-exp trick. Note that the earlier trick cannot be directly applied to l^t due to the sum $\sum_{k=1}^{K}$ being inside the $\log$. Additionally, we must consider the trade-off between the stable upper bound and l^t (referred to as the stabilization margin), as a larger gap between them implies a greater misalignment between the training and inference procedures. Thus, instead of minimizing l^t, we minimize the quantity presented in line 14 of Algorithm 1. Here's a detailed explanation:

Due to the inclusion of the sum $\sum_k$ within the logarithm, we cannot directly apply the log-sum-exp trick to stabilize it. We want instead to bound this quantity by the tightest upper bound term, which can be stabilized using the log-sum-exp trick.

$$l^t = -\log(\sum_{k} d_k(\sum_{i} p_{i,j_t^*}^k))) \tag{5.10}$$

$$= -\log(\sum_{k} d_k(\sum_{i} \frac{\exp(\frac{-w_{i,j_t^*}^k}{\alpha})}{\sum_{i,j_t^*} \exp(\frac{-w_{i,j_t^*}^k}{\alpha})})) \tag{5.11}$$

Using the concavity of the $\log$ function, and given that d_k is a prior over the set of beams, such that $\sum_k d_k = 1$, we can establish the following inequality:

$$l^t \leq \sum_{k} d_k(-\log(\sum_{i} \frac{\exp(\frac{-w_{i,j_t^*}^k}{\alpha})}{\sum_{i,j_t^*} \exp(\frac{-w_{i,j_t^*}^k}{\alpha})})) \tag{5.12}$$

$$= \sum_{k} d_k(-\log(\frac{\sum_i \exp(\frac{-w_{i,j_t^*}^k}{\alpha})}{\sum_{i,j_t^*} \exp(\frac{-w_{i,j_t^*}^k}{\alpha})})) \tag{5.13}$$

Algorithm 1 Soft SLBS training for IE

```
Input: $x = x^1, \dots, x^T$, the linearized graph
for t=1 to T do
    for i=1 to K do
        $h_i^t \leftarrow \text{RNN}(x^{t+1}, \mathcal{E}(\hat{c}^t_{\cdot,i}), h_i^t)$
        for j=1 to $|\mathcal{V}_{task}|$ do
            $\hat{y}^t_{i,j} \leftarrow \text{FFN}_{task}(h_i^t)_j$
            $\tilde{s}^t_{i,j} \leftarrow s_i^t + \hat{y}^t_{i,j}$
        for j=1 to $|\mathcal{V}_{task}|$, k=1 to K do
            $w^k_{i,j} \leftarrow (\tilde{s}^t_{i,j} - \underset{\substack{1 \le i \le K \\ 1 \le j \le |\mathcal{V}_{task}|}}{\texttt{top-k-max}}(\tilde{s}^t_{i,j}))^2$
            $p^k_{i,j} \leftarrow \sigma(\frac{-w^k_{\cdot,\cdot}}{\alpha})_{i,j}$
        for k=1 to K do
            $s_k^{t+1} \leftarrow \sum_{i,j} p^k_{i,j} \tilde{s}^t_{i,j}$
        for j=1 to $|\mathcal{V}_{task}|$, k=1 to K do
            $\hat{c}^t_{j,k} \leftarrow \frac{\sum_i p^k_{i,j}}{i}$
    $loss+ = \sum_k d_k((-\log(\sum_i e^{-w^k_{i,j^*_t}+a}) + a) + (\log(\sum_{i,j^*_t} e^{-w^k_{i,j^*_t}+b}) - b)$
    with $a = \min_i(w^k_{i,j^*_t})$ and $b = \min_{i,j}(w^k_{i,j^*_t})$
```

Next, by applying the log-sum-exp trick to the obtained upper bound, we can derive a stable upper bound for l^t:

$$l^t \le \sum_k d_k((-\log(\sum_i e^{-w^k_{i,j^*_t}+a}) + a) \tag{5.14}$$

$$+ (log(\sum_{i,j^*_t} e^{-w^k_{i,j^*_t}+b}) - b) \tag{5.15}$$

with $a = \underset{i}{min}(w^k_{i,j^*_t})$ and $b = \underset{i,j}{min}(w^k_{i,j^*_t})$.

Total Loss The model is jointly trained to minimize the nodes identification loss and the labeled graph generation loss: $L = L_{id} + L_{cl}$.

Dataset	Split	SENT	ENT	REL	EVT	ARG
ACE05-R	Train	10,051	26,473	4,788	-	-
	Dev	2,424	6,338	1,131	-	-
	Test	2,050	5,476	1,151	-	-
CoNLL04	Train	922	3,377	1,283	-	-
	Dev	231	893	343	-	-
	Test	288	422	422	-	-
ACE05-E	Train	19,240	47,554	7,159	4,419	6,607
	Dev	901	3,423	728	468	759
	Test	676	3,673	802	424	689
ACE05-CN	Train	6,841	29,657	7,934	2,926	5,463
	Dev	526	2,250	596	217	403
	Test	547	2,388	672	190	332
ACE05-AR	Train	2,936	26,031	3,712	1,830	3,176
	Dev	382	3,256	498	234	401
	Test	371	2,925	392	204	334

Table 5.1: Statistics of the Used ACE05 and CoNLL04 Datasets.

5.4 Experimental Setup

5.4.1 Datasets

We evaluate our model on 2 datasets and 3 different languages: ACE05 (Walker and Consortium, 2005) for English, Arabic, and Chinese, and CoNLL04 Roth and Yih (2004). For English ACE05, we consider two versions from the literature: ACE05-R, which involves entity and relation extraction, and ACE05-E+, which includes entity, relation, and event extraction. We follow the data splits and preprocessing of Luan et al. (2019) and Lin et al. (2020) for ACE05-R and ACE05-E+. For Chinese data, we use the same preprocessing and splits of Lin et al. (2020) and refer to it by ACE05-CN. For Arabic data, we use the same preprocessing and splits of El Khbir et al. (2022) and refer to it by ACE05-AR. Thus, CoNLL04 involves 4 entity types and 5 relation types, and ACE05 involves 7 entity types, 6 relation types, 33 event types, and 22 argument types. Table 5.1 provides statistics of the datasets, where **SENT**, **ENT**, **REL**, **EVT**, **ARG** denotes respectively the number of sentences, entities, relations, event triggers and event arguments.

5.4.2 Evaluation Metrics

We use micro F1 measure for evaluation. Entity and event trigger predictions are correct when the type and boundaries match the gold data. For relations and event arguments, we adopt boundaries evaluation (Taillé et al., 2020), a nonstrict and undirected evaluation. A relation or an argument is correct if the type and boundaries match the gold data. Additionally, we report average F1 scores (AVG) across all tasks for global assessment. We average scores from 3 runs and select the model with the highest average F1 on the dev set.

5.4.3 Settings and Hyperparameters

Pretrained Language Model For the PLMs, we use *bert-large-cased* Devlin et al. (2019) for the ConLL04, ACE05-R, and ACE05-E+ datasets, *bert-large-arabertv2* Antoun et al. (2020) for the ACE05-AR dataset, and *bert-large-chinese* for the ACE05-CN dataset. We fine-tune the hyperparameters on ACE05-E+ and apply the same settings to the other ACE05 datasets.

Hyperparameter Tuning We perform a hyperparameter search for the beam size (K) with values in $\{4, 10, 16, 20, 22\}$ and select $K = 10$ based on performance. For the temperature parameter (α), we explore values in $\{0.1, 0.5, 1, 2, 5, 10\}$ and retain α=1. We use a uniform prior for d_k. Additional hyperparameters used in our experiments include the Adam optimizer, with a BERT learning rate of 1e-5, BERT weight decay of 1e-5, and BERT dropout of 0.5. We also implement gradient clipping at 5.0, a learning rate of 1e-4, weight decay of 1e-4, and dropout of 0.4. The hidden sizes are set to 256 for the RNN, 150 for FFN$_{node}$, and 600 for FFN$_{edge}$.

Computaional Resources Our experiments run on an Nvidia GEForce RTX 2080 GPU with 8 GB of RAM. The estimated computational budget for each training epoch is approximately 3 GPU minutes for ConLL04, 10 GPU minutes for ACE05-R, 20 GPU minutes for ACE05-E+, 6 GPU minutes for ACE05-AR, and 5 GPU minutes for ACE05-CN.

5.5 Results and Analysis

Main Results The main results of our experiments on ConLL04 and ACE05 data, along with some literature results, are presented in Tables 5.2 and 5.3. We begin by establishing a baseline with the Sequence Labeling Beam Search model (SLBS), trained through a greedy approach and decoded using beam search (§5.3.3). This foundational model provides a crucial benchmark against which subsequent enhancements are evaluated.

We then present the results of the Soft Sequence Labeling Beam Search (SSLBS) model, trained with a relaxed beam search strategy and decoded using beam search (§5.3.4).

The results show that the SSLBS model outperforms the baseline, as evidenced by the average F1 score enhancement across all used datasets. Specifically, the SSLBS model demonstrates enhancements of 1.4, 0.3, 0.8, 2.0, and 1.1 F1 score points on the ConLL04, ACE05-R, ACE05-E+, ACE05-CN, and ACE05-AR datasets, respectively. This observation

Model	CoNLL04			ACE05-R			ACE05-E+				
	ENT	**REL**	**AVG**	**ENT**	**REL**	**AVG**	**ENT**	**REL**	**EVT**	**ARG**	**AVG**
Wang and Lu (2020)$^{\times}$	**90.1**	**73.8**	**81.9**	89.5	67.6	78.5	-	-	-	-	-
Wadden et al. (2019)* +	-	-	-	88.4	63.2	75.8	-	-	-	-	-
Zhong and Chen (2021)*	-	-	-	88.7	66.7	77.7	-	-	-	-	-
Ye et al. (2022)*	-	-	-	**89.8**	69.0	**79.4**	-	-	-	-	-
Zhang and Ji (2021)†	-	-		88.7	67.2	77.9	91.0	62.8	72.7	57.7	71.0
Nguyen et al. (2022b)†	-	-	-	-	-	-	**91.7**	64.9	74.6	**61.2**	**73.1**
Lin et al. (2020)$^{\diamond}$	-	-	-	88.8	67.5	78.1	89.6	58.6	72.8	54.8	69.0
Nguyen et al. (2021a)$^{\diamond}$	-	-	-	88.9	68.9	78.9	91.1	63.6	73.3	57.5	71.4
Nguyen et al. (2022a)$^{\diamond}$	-	-	-	88.9	**69.5**	79.2	91.0	**65.4**	74.8	59.9	72.7
SLBS $^{\diamond}$	90.0	68.6	79.4	88.9	68.2	78.6	91.4	63.8	73.3	55.6	71.0
SSLBS $^{\diamond}$	90.1	71.4	80.8	88.5	69.2	78.9	91.2	64.0	**75.0**	56.9	71.8

Table 5.2: Performance on English. Models grouped in the same group of rows use the same encoder for word representations; ×: *albert-xxlarge*, ∗: *bert-base*, †: *roberta-large*, ⋄: *bert-large*. Models marked with a + sign use extra training data.

		ACE05-CN			
Model	**ENT**	**REL**	**EVT**	**ARG**	**AVG**
Lin et al. (2020)$^{\diamond}$	88.5	62.4	65.6	52.0	67.1
Nguyen et al. (2021a)$^{\diamond}$	88.7	65.1	66.5	54.9	68.8
Nguyen et al. (2022b)*	**89.2**	**68.3**	**74.3**	**60.0**	**72.9**
SLBS†	88.6	64.8	65.9	49.6	67.3
SSLBS†	**89.2**	67.1	68.3	52.4	69.3

		ACE05-AR			
Model	**ENT**	**REL**	**EVT**	**ARG**	**AVG**
El Khbir et al. (2022)$^{\times}$	85.1	62.9	63.6	51.8	66.0
SLBS$^{\times}$	**85.3**	**63.1**	62.0	51.6	65.5
SSLBS$^{\times}$	84.6	**63.1**	**63.9**	**55.0**	**66.6**

Table 5.3: Performance on Chinese and Arabic. ⋄:*bert-multilingual-cased*, ∗:*xlm-roberta-large*, †:*bert-large-chinese*, ×:*bert-large-arabertv2*

strongly indicates the superiority of the decoding-aware training strategy over traditional greedy training methods.

Comparison to other works For English, we compare our model to Lin et al. (2020), Nguyen et al. (2021a), and Nguyen et al. (2022a) since we use the same PLM as an encoder. Among these works, SSLBS has the second-best relation and average F1 scores on ACE05-

R, the best trigger F1 score, and the second-best entity, relation, and average F1 score on ACE05-E+. In addition, we consider other joint IE models such as Wadden et al. (2019); Zhang and Ji (2021); Nguyen et al. (2022b), as well as models that focus solely on joint ERE Wang and Lu (2020); Zhong and Chen (2021); Ye et al. (2022). While these models employ various techniques such as span graph propagation Wadden et al. (2019), manually-designed global features Lin et al. (2020); Zhang and Ji (2021), global type dependency regularization Nguyen et al. (2021a), and dependency-induced graphs with simulated annealing Nguyen et al. (2022a), the SSLBS model implicitly learns graph representations through the hidden states of the network.

For Arabic and Chinese, SSLBS exhibits comparable performance to other existing approaches, with the trigger and argument tasks showcasing substantial performance gains.

Overall, while SSLBS does not surpass all SOTA models, it still achieves competitive scores. To ensure fairness in comparisons, evaluating with the same PLM is preferable Taillé et al. (2020). However, the focus of our work is on integrating the decoding procedure into training, rather than exploring different PLM parameters. We make our code publicly available for further investigations.

5.6 Ablation Studies

5.6.1 Effect of Forward/ Prediction Beam Sizes

To ensure alignment between training and inference objectives, we investigate the impact of different beam sizes on our model's performance. We denote here *fbs*, the forward beam size used during training, and *pbs* the prediction beam size used during inference. Figure 5.3 shows the obtained average F1 scores, with a fixed temperature α=1, for ACE05-E+ dataset.

We notice that the diagonal of the matrix, corresponding to *fbs*=*pbs*, is prevailing. This indicates that the model achieves its best results when the training closely aligns with the inference process.

In addition, we notice that the scores of the over-diagonal, corresponding to *fbs* > *pbs*, consistently outperform those of the under-diagonal, corresponding to *fbs* < *pbs*. This suggests that a model trained with a larger beam size has a broader exposure to potential options during training, enabling it to better handle search errors that occur when decoding with a smaller beam size. Conversely, the lowest score is obtained for the $\{fbs = 10, pbs = 22\}$ combination, which highlights a performance decline when the beam size used during decoding is larger than that during training. These insights emphasize the importance of aligning beam sizes to enhance model performance and generalization.

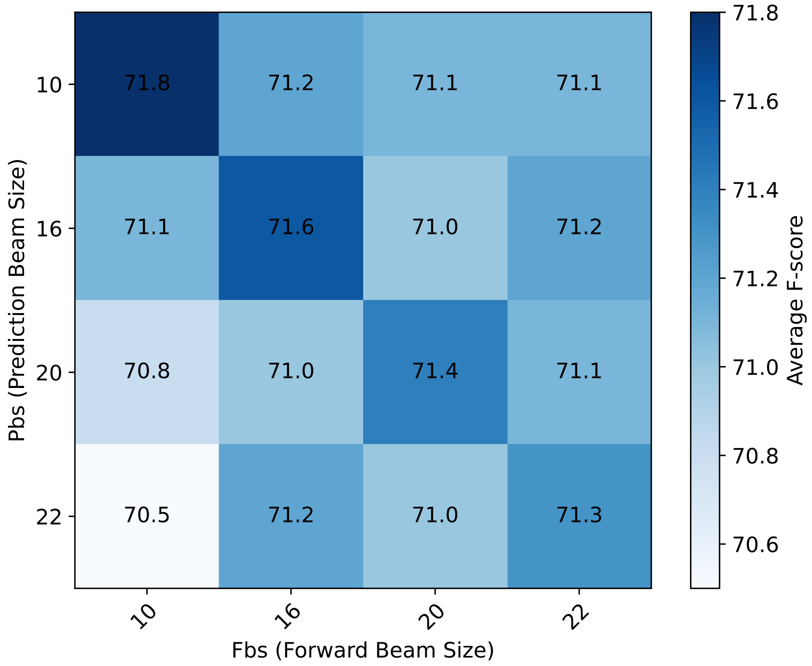


Figure 5.3: Effect of *fbs* and *pbs* on performance - ACE05-E+ Data.

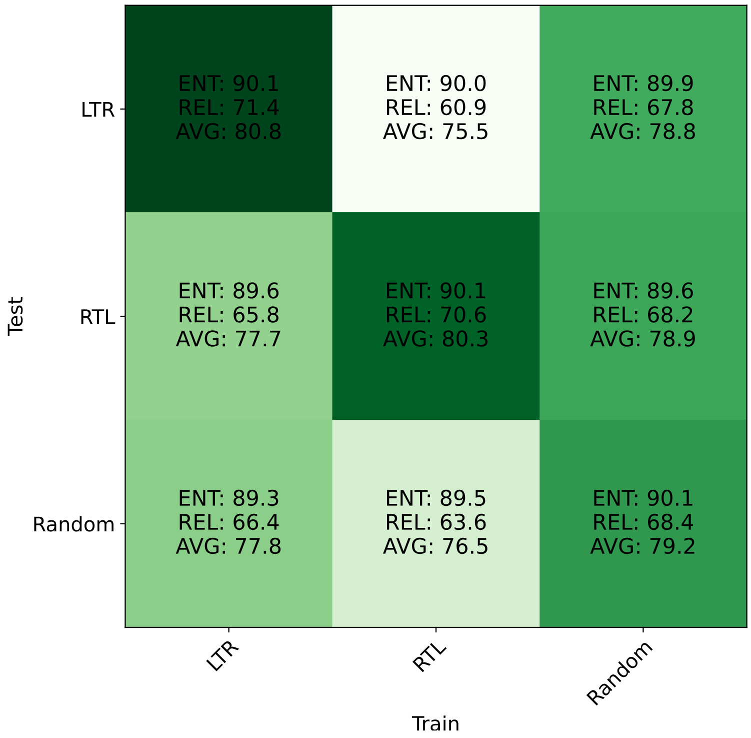


Figure 5.4: Effect of Sequence Ordering on Performance - CoNLL04 Data.

5.6.2 Effect of Sequence Ordering

We perform experiments to explore the impact of varying sequence orders during both the training and testing phases. For all previous experiments, we have adhered to the sequence

MODEL	TF					no TF				
	SLBS	SSLBS	SSLBS	SSLBS	SSLBS	SLBS	SSLBS	SSLBS	SSLBS	SSLBS
		$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$		$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$
EB	258.7	146.2	142.4	25.7	82.3	163.3	544.6	39.6	3.7	112.0
FVC	192	8	15	21	30	35	15	13	15	22
ENT	89.4	90.2	90.1	90.3	89.9	90.1	90.0	90.4	90.3	89.5
REL	69.5	66.7	68.3	71.3	70.3	69.2	67.3	68.3	71.3	67.6
AVG	79.4	78.4	79.2	80.8	80.1	79.6	78.6	79.3	80.8	78.6

Table 5.4: Exposure Bias Quantification.

order outlined in §5.3.2, denoted here as the left-to-right (LTR) order. However, to comprehensively assess our model's performance, we introduce two alternative sequence orders: the right-to-left (RTL) order and a random (Random) order. In the RTL order, we maintain fixed node positions while rearranging the edges in a right-to-left fashion. Conversely, the Random order involves a random reordering of edges while keeping node positions constant. We conduct these experiments on ConLL04, and the results are depicted in Figure 5.4.

We observe a dominant trend along the diagonal in Figure 5.4, which indicates that the model consistently excels when tested on the same order it was trained on, thus when training and inference are aligned. Notably, training and testing with the LTR ordering consistently yield the best performance, possibly because the LTR order aligns well with the natural sequential dependencies of the data.

Additionally, training the model with the Random order and testing it with different orders (last column) demonstrates superior adaptability and robustness compared to training with either LTR or RTL. The model's ability to adapt to novel sequence arrangements stands out in this scenario.

5.6.3 Exposure Bias Quantification

We assess exposure bias in two settings: SLBS and SSLBS (with various temperature α values). We also explore the use of Teacher Forcing (TF) and model predictions (no TF) in both settings. We conduct these experiments on ConLL04, training the model for 150 epochs and reporting results of the last epoch in Table 5.4.

Exposure bias refers to the gap between a model's training and testing conditions. We quantify exposure bias by computing the Kullback-Leibler divergence between the distributions of training hidden states $P_{h_{\text{train}}}$ and decoding hidden states $P_{h_{\text{test}}}$. We practically compute this using an N-samples Monte-Carlo scheme:

$$D_{\text{KL}}(P_{h_{\text{train}}} \,||\, P_{h_{\text{test}}}) \underset{h_i \sim P_{h_{\text{train}}}}{\approx} \frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{P_{h_{\text{train}}}(h_i)}{P_{h_{\text{test}}}(h_i)} \right) \tag{5.16}$$

Besides, we approximate these hidden state distributions $P_{h_{\text{train}}}$ and $P_{h_{\text{test}}}$ as Gaussian

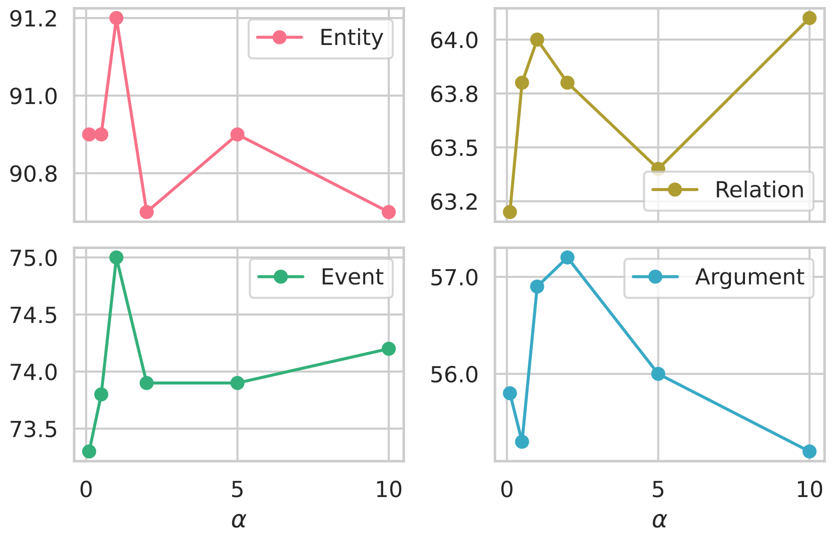


Figure 5.5: Effect of Temperature on Performance - SSLBS - ACE05-E+ Data.

Mixtures Reynolds (2009), using 5 components.

Practically, the trained models make little to no use of certain hidden dimensions. To streamline calculations and reduce noise in hidden states, we employ Principal Component Analysis (PCA) F.R.S. (1901) to retain the principal components explaining 95% of the variance in training hidden states. These dimensionally reduced hidden states are then used to fit GMMs approximating $P_{h_{\text{train}}}$ and $P_{h_{\text{test}}}$. Note that the number of principal components required to explain 95% of the variance in training hidden states serves as a measure of the vectorial complexity of a model's hidden states. In this context, these states inhabit a lower-dimensional hyperplane than that of the latent space. This measure, which we call Features Vectorial Complexity (FVC), is reported in Table 5.4, alongside the exposure bias values (EB), the application of teacher forcing (TF/ no TF), and the F1 scores for entities (ENT), relations (REL), and their average (AVG).

In Table 5.4, we observe that an increase in exposure bias is associated with lower F1 scores. We compute the Spearman correlation between performance (AVG) and exposure bias (EB) values for all models, as well as specifically for SSLBS models. The resulting correlation coefficients are -0.59%, and -89% indicating a robust negative association between these two variables, which validates our initial observation, highlighting the adverse effect of exposure bias on performance.

5.6.4 Effect of the Temperature parameter

We conducted experiments on ACE05-E+ varying the temperature parameter α in the range $\{0.1, 0.5, 1, 2, 5, 10\}$ to study its impact on performance. As shown in Figure 5.5, the model with an intermediate temperature ($\alpha = 1$) achieved the highest performance, indicating better training stability and model confidence calibration.

5.7 Related Work

Prior research has approached entity recognition (ER) and relation extraction (RE) as separate tasks Zhou and Su (2002); Zelenko et al. (2002); Kambhatla (2004), and others addressed both entity and relation extraction (ERE) tasks jointly Chan and Roth (2011); Yu et al. (2019). Recent works address the four tasks of entity, relation, trigger, and argument extraction jointly Luan et al. (2019); Wadden et al. (2019); Lin et al. (2020); Zhang and Ji (2021); Nguyen et al. (2022b).

Seq2Seq Models Some works proposed Sequence-to-Sequence architectures for ERE. While Miwa and Bansal (2016) used an encoder-decoder architecture with attention, they relied on expensive trees. In contrast, Yu et al. (2019) reformulated ERE as a single sequence labeling task but did not handle overlapping relations effectively. To our knowledge, we are the first to recast the four tasks as a joint sequence labeling problem.

Exposure Bias Solutions Exposure bias is a known issue in Seq2Seq models across various NLP domains. Solutions include reinforcement learning models (Ranzato et al., 2016), beam search training schemes with sequence-level cost functions (Wiseman and Rush, 2016), and differentiable relaxations of beam search procedures Goyal et al. (2018). These methods have been applied to tasks such as NER and CCG Supertagging, demonstrating their effectiveness. Many researchers have tried to address this issue using various techniques:

- **Schedule sampling**: This method gradually replaces reference tokens with model predictions as training progresses, forcing the model to become more reliant on its own predictions and reducing the mismatch between training and inference.
- **Curriculum learning**: This technique starts with easier prediction tasks and gradually increases the difficulty, allowing the model to learn effective strategies before tackling more complex scenarios.
- **Knowledge distillation**: This involves training a smaller, faster model to mimic the predictions of a larger, more accurate model. By focusing on replicating the "good" decisions of the large model, the smaller model can be less susceptible to exposure bias.
- **Auxiliary losses**: These additional losses encourage the model to attend to specific aspects of the input or produce intermediate outputs that are closer to the desired labels, guiding the model's learning process and reducing the impact of exposure bias.

Our work Our approach, similar to that of You et al. (2018), uses linearization to transform graph structures into sequential representations. However, while You et al. (2018) explicitly models the generative process, our work centers on predicting graph-related tasks. Our continuous beam search procedure, inspired by Goyal et al. (2018), integrates four tasks into the procedure, optimizes a task-specific loss, and utilizes a straightforward RNN recurrence to implicitly combine contributions from various beam elements for computing subsequent steps.

5.8 Conclusion and Discussion

In this work, we addressed the challenge of information extraction graph generation. We proposed a two-step approach:

- **Sequence Labeling with Beam Search (SLBS)**: We first reformulated the problem as a sequence labeling task using autoregressive models. During decoding, we employed beam search to improve performance by considering a set of top candidate sequences. This initial model, however, relied on a greedy selection process during beam search, limiting its learning potential.
- **Soft Sequence Labeling with Beam Search (SSLBS)**: To overcome this limitation, we introduced SSLBS. Unlike SLBS, SSLBS leverages a differentiable beam search approach during training. This key innovation allows the model to learn from all candidate sequences within the beam, not just the single, greedily chosen one. This richer learning process enables SSLBS to achieve superior performance compared to the baseline SLBS model.

Through ablation studies, we gained valuable insights into factors influencing model performance:

- **Beam Size Alignment**: We observed that aligning beam sizes during training and inference leads to optimal performance. This suggests that the model benefits from consistency in the search space during both phases. However, using a larger training beam with a smaller inference beam size demonstrates some model adaptability. This opens possibilities for exploring strategies to reduce computational cost at inference time while maintaining accuracy.
- **Sequence Ordering**: The model performs best when the training and testing order of entity and relation sequences are aligned. However, training with random order shows a degree of adaptability. This suggests that the model can learn to handle different ordering scenarios to some extent. Future work could investigate methods to further improve this adaptability and make the model less sensitive to ordering variations in real-world data.
- **Exposure Bias**: The strong negative correlation between exposure bias and F1 scores highlights the importance of mitigating this challenge, where models struggle to generalize when trained and tested under different conditions. SSLBS demonstrates an advantage in this regard, potentially due to its differentiable formulation. Exploring additional techniques to reduce exposure bias in joint information extraction tasks could be a fruitful direction for future research, leading to models that are more generalizable and robust to real-world data variations.

- **Temperature Parameter**: The optimal performance achieved with an intermediate temperature value suggests good training stability and model calibration in SSLBS. This parameter controls the exploration-exploitation trade-off during beam search. Further investigation into the role of the temperature parameter in different information extraction settings could provide valuable insights for model tuning, potentially leading to improved performance across various datasets and tasks.

This work opens new avenues for exploring differentiable beam search techniques in information extraction tasks. The model demonstrates promising results, outperforming baselines and achieving competitive scores. However, this work also acknowledges some limitations. Future research can address these limitations by:

- Exploring strategies for reducing computational cost at inference: While SSLBS demonstrates strong performance, beam search can be computationally expensive. Investigating techniques to reduce the beam size or exploring approximate inference methods could make the model more practical for large-scale deployments.
- Addressing vanishing gradients with decreasing temperature: The temperature parameter α controls the model's confidence in its predictions. However, as α decreases, the model can experience vanishing gradients, hindering training. This creates a trade-off between the stability and the accuracy of the predictions.
- Improving model adaptability to sequence ordering: While SSLBS shows some adaptability to different ordering scenarios, further improvements in this area involve making the model learn such an ordering itself.

Chapter 6

Conclusion

In this thesis, we have explored various facets of information extraction, addressing several core tasks: entity extraction (EE), relation extraction (RE), event trigger extraction (ETE), and event argument extraction (EAE). Our primary focus was on the Arabic language, given its rich linguistic nuances and complexities. By addressing its particularities, we aimed to enhance the efficiency and accuracy of IE tasks through the development of novel methodologies specifically tailored to the challenges posed by Arabic and its diverse dialects.

The body of work presented in this thesis spans three significant contributions, each tackling distinct challenges and introducing innovative solutions to advance the field of information extraction. In Chapter 3, we presented *ArabIE*, the first neural joint IE model designed for Modern standard Arabic (MSA), addressing the intricacies of Arabic morphology and syntax. In Chapter 4, we focused on cross-dialectal named entity recognition (NER), leveraging cross-lingual transfer learning to extend IE capabilities to various Arabic dialects. Finally, in Chapter 5, we introduced a novel approach to IE using differentiable beam search on graph recurrent neural networks to model the interdependencies between different IE tasks more effectively. This last work extends beyond Arabic-specific models, demonstrating broader applicability in general information extraction tasks across multiple languages.

In the following sections, we summarize the key findings, implications, and future directions of each work, reflecting on the advancements, limitations, and impact of our contributions to information extraction for Arabic and beyond.

6.1 Joint Entity, Relation, and Event Extraction for Arabic

6.1.1 Objectives and Achievements

The first objective of this thesis was to develop a robust joint IE model for Modern Standard Arabic, addressing the unique morphological and syntactic challenges of the language. To achieve this, we introduced *ArabIE*, the first neural joint IE model for Arabic. *ArabIE* simultaneously tackles four essential tasks: named entity recognition, relation extraction, event trigger extraction, and event argument extraction.

ArabIE leverages BERT as a token encoder and uses two models trained in a multitask fashion. Conditional Random Fields (CRFs) are used for node identification, encompassing entities and triggers, while feed-forward networks (FFNs) are used for both node and edge classification, encomassing entities, relations, triggers, and arguments classification. To handle Arabic's complex morphology, we explored two tokenization approaches during the preprocessing steps: morphological tokenization and augmented BIO tags. Our key results and findings include:

- **Comparable Results to SOTA models**: *ArabIE* demonstrated good performance on the ACE 2005 benchmark, achieving results comparable to leading models for other languages. Specifically, *ArabIE* achieved an average of 65.84 F1 points on all tasks when using the morphological tokenization approach. In comparison, state-of-the-art models have achieved 68.95 F1 points for English, 67.12 for Chinese, and 56.62 for Spanish on the corresponding ACE datasets. This performance is significant for the first Arabic information extraction model, highlighting its competitive edge.
- **Morphological Tokenizer Superiority**: The morphological tokenizer, which segments words into morphemes, proved to be the most effective approach, achieving a performance increase of 2 F1 points compared to other tokenization methods. This approach preserved valuable subword information better than other techniques. Conversely, the augmented BIO tags method, which concatenates labels for subword entities, resulted in an increased label set size and was less effective.
- **Impact of Annotation Omissions**: Despite the ACE 2005 dataset being of high quality and granularity, error analysis revealed significant discrepancies in annotation quality within the gold standard data. In a sample of 32 sentences, nearly 23.5% contained annotation errors, particularly concerning triggers and roles. The model often predicted correct events according to the annotation guidelines but was penalized due to omissions by annotators. These inconsistencies posed challenges for model training and evaluation, highlighting the need for robust evaluation metrics beyond traditional accuracy measures.

6.1.2 Limitations

Despite its success, *ArabIE* faces some limitations, including:

- **Random Entity Selection**: After tokenizing the words, projecting entities onto tokens is not always perfect due to inconsistencies between the morphemes and entities of the gold data, or the tokenizer failing to produce valid morphemes. This issue can lead to significant data loss, with an estimated 1% of data being affected. When a subword still holds multiple entities after tokenization, the model randomly selects one entity and discards the others. This exclusion of relevant information impacts the model's overall accuracy. Future research should focus on developing more sophisticated methods for entity selection during tokenization to mitigate data loss. For instance, exploring character-based tokenization might provide a more granular approach, allowing the model to capture multiple entities within a single token more effectively.

- **Tokenization-Vocabulary Mismatch**: The mismatch between the vocabularies generated by tokenizers and the BERT model used for token encoding can significantly affect performance. This discrepancy occurs because the tokenizer's output may not perfectly align with the BERT vocabulary, resulting in suboptimal token representations and potentially reducing the accuracy and effectiveness of the model. Training a custom BERT model specifically on the output of the chosen tokenizer could address this issue, ensuring that the token representations are more aligned and accurate.
- **Limited Inter-Task Communication**: There is restricted communication between tasks in the model due to the employed greedy decoding strategy, which does not explicitly account for task dependencies. Effective inter-task communication is crucial for capturing relationships among entities, relations, and events. The lack of such communication can result in isolated predictions, reducing overall coherence and accuracy. In Chapter 5, we addressed this by incorporating techniques that consider these dependencies, leading to significant performance improvements.

6.1.3 Future Directions

This work establishes a robust baseline for Arabic information extraction systems, showcasing notable performance across four critical tasks: named entity recognition, relation extraction, event trigger extraction, and event argument extraction. The methodologies and models developed here are versatile and can be directly applied or adapted for various real-world applications. For instance, in automated news aggregation, these models can streamline compiling and categorizing news articles by accurately identifying key entities and events. In social media monitoring, extracting nuanced relationships and event triggers can significantly enhance the understanding of public sentiment and trends.

For the Arabic NLP research community, this work serves as a critical foundation for further exploration. Researchers can leverage the findings and models presented here to develop more advanced systems. The detailed error analysis and insights into the impact of tokenization strategies offer valuable directions for future research. Understanding where current models fall short allows to target specific areas for improvement, such as refining tokenization methods to better capture the complexities of Arabic morphology or developing more sophisticated annotation techniques to minimize errors. These insights can drive the creation of more accurate and reliable IE systems.

6.2 Cross-Dialectal Named Entity Recognition in Arabic

6.2.1 Objectives and Achievements

The second objective was to extend IE capabilities to Arabic dialects, which are widely used in everyday communication but pose significant challenges due to their linguistic diversity. To this end, we conducted a comprehensive study on cross-dialectal named entity recognition, leveraging cross-lingual transfer learning from MSA to Arabic dialects. This approach aimed to bridge the linguistic gap and enable effective entity extraction across different dialects.

To extend NER capabilities to Arabic dialects, we employed a span-based NER model built on top of a pretrained language model (PLM) encoder. This model was trained on MSA data and tested on dialectal data to leverage cross-lingual transfer learning. The process began with the manual annotation of NER datasets for Moroccan, Egyptian, and Syrian dialects, chosen to capture the broad linguistic diversity within the Arabic-speaking world. These dialects span North Africa, the Middle East, and the Levant, and Egyptian and Syrian Arabic are widely understood due to cultural influence. The PLM encoder, fine-tuned on MSA data, provided robust contextual embeddings that captured the linguistic nuances of the text. The span-based model then identified and classified named entities within these embeddings. Evaluation was conducted using standard metrics such as precision, recall, and F1 score to measure the model's performance across different dialects. Our key contributions and findings include:

- **Creation of Manually Annotated NER Datasets**: We developed NER datasets for Moroccan, Egyptian, and Syrian dialects through manual annotation, adhering to high-quality LDC guidelines. These datasets have been made publicly available for future research.

- **Effectiveness of Cross-Lingual Transfer Learning**: The model demonstrated strong generalization capabilities to different dialects, particularly Syrian Arabic, achieveing the highest F1 scores across various PLMs. Using AraBERTv2 as the PLM, the model trained on MSA achieved F1 scores of 59.44 for Egyptian, 55.23 for Moroccan, and 66.97 for Syrian dialects. This success in zero-shot settings highlights the potential of cross-lingual transfer learning to overcome data scarcity and deliver high performance in dialectal NER tasks.

- **Linguistic Affinity Findings**: Our results revealed that Syrian Arabic has the closest linguistic affinity to MSA, resulting in higher NER performance across all tested PLMs, followed by Egyptian and then Moroccan dialects.

6.2.2 Limitations

This preliminary study presents a good starting point for dialectal NER but faces some limitations, including:

- **Single Annotator Bias**: The dataset relies on a single annotator, which may introduce bias and affect the reliability of the labels. Single annotator bias can lead to subjective interpretations influencing the consistency and accuracy of annotations. Future work should consider the involvement of multiple annotators to assess inter-annotator agreement and ensure labeling robustness.
- **Limited Dialect Scope**: Our work was limited to only three Arabic dialects: Moroccan, Egyptian, and Syrian. This narrow focus does not capture the full linguistic diversity of Arabic, which includes many other dialects with unique features. A more comprehensive study across multiple dialects would provide a broader understanding of the challenges and solutions in Arabic NER.

Besides these two limitations, the dataset created for this work is relatively small, comprising a total of 1,592 annotated sentences across MSA and the three dialects, and a total of 5,787 annotated entities. While zero-shot transfer learning demonstrated decent performance, the limited size of the dataset could hinder the model's generalizability to other applications, particularly those involving domain-specific entities. A larger and more diverse dataset would enable better training and evaluation, enhancing the robustness and applicability of the model.

6.2.3 Future Directions

As stated before, this preliminary work shows significant potential for advancing dialectal IE. The model can be directly used for general domain NER on written MSA, Egyptian, Moroccan, and Syrian dialects, benefiting both research and practical applications, such as extracting information from newspapers, social media, and other text sources. However, real-life communication in dialects presents several challenges that need further study and innovative solutions.

Many social media users do not write their everyday communications in Arabic script but use Latin letters to phonetically represent Arabic words, known as “Arabizi”. This poses a unique challenge for NER models trained on Arabic script. Future research should focus on developing models that can handle Arabizi by incorporating character-level embeddings and creating annotated datasets in this script. This adaptation will make the model applicable to a broader range of text inputs.

Another challenge in real-life dialectal Arabic communication is code-switching, where speakers mix words from different languages within the same sentence. For instance, Egyptian dialect often includes English words, while Moroccan dialects may mix French and English. Creating models to manage code-switching is a key research area, requiring strategies to process multiple languages within a single sentence.

A promising direction is integrating advanced LLMs like GPT-4, via prompting or fine-tuning, to enhance the performance and generalization capabilities of the NER model. These models can capture more nuanced linguistic features and better handle the complexities of dialectal and code-switched texts, providing more accurate and context-aware entity recognition.

6.3 Information Extraction with Differentiable Beam Search on Graph RNNs

6.3.1 Objectives and Achievements

The third major objective of this thesis was to model the interdependencies between different IE tasks more effectively. To achieve this, we employed autoregressive models and beam search techniques, developing a novel approach that frames information extraction as a sequence labeling problem, using autoregressive models to label a linearized IE graph.

We introduced two models in this work: Sequence Labeling with Beam Search (*SLBS*) and Soft Sequence Labeling with Beam Search (*SSLBS*). *SLBS* leverages a linear-chain Conditional Random Field (CRF) with a BIO tagging scheme for node identification (entities and triggers) and uses an auto-regressive Recurrent Neural Network (RNN) with beam search decoding for labeled graph generation. This model is trained in a greedy way using teacher forcing but decoded with a beam search strategy, which induces exposure bias, or train-eval inconsistency, leading to reduced performance. To address this issue, *SSLBS* aligns training and inference using a differentiable beam search procedure during training, allowing the model to account for its decoding behavior during training, thus mitigating exposure bias. Our key findings and contributions include:

- **Alignment of Training and Inference**: By integrating a differentiable beam search into the training process, *SSLBS* aligns the training and inference procedures, significantly improving model performance. The effectiveness of our model was validated through extensive experiments on the ACE05 dataset (covering English, Arabic, and Chinese) and the CoNLL04 dataset. The results demonstrated that *SSLBS* significantly outperforms *SLBS* in terms of F1 scores across all datasets. Specifically, The *SSLBS* model demonstrated improvements of 1.4, 0.3, 0.8, 2.0, and 1.1 on average F1 score points on CoNLL04, ACE05-R, ACE05-E+, ACE05-CN, and ACE05-AR, respectively.
- **Ablation Studies**: We conducted ablation studies to examine the impact of aligning training and evaluation processes on model performance. Key findings include:
  - **Beam Size Alignment**: Using the same beam size for both training and inference led to optimal performance. This alignment ensured consistency and reduced discrepancies between the two stages, highlighting the importance of maintaining identical conditions during both phases.
  - **Sequence Ordering**: Maintaining a consistent sequence order during both training and testing enhanced model performance. Training with random orders demonstrated the model's adaptability to different scenarios but resulted in slightly reduced performance compared to consistent ordering.
- **Exposure Bias Quantification**: We quantified exposure bias by computing the Kullback-Leibler divergence between the distributions of training hidden states and decoding

hidden states. Lower exposure bias values correlated strongly with higher F1 scores, underscoring the negative impact of train-eval discrepancies.

6.3.2 Limitations

Despite the advancements made with the *SSLBS* model, several limitations persist, including technical limitations such as vanishing gradients. The use of temperature parameters in the continuous relaxation of beam search helps control the model's confidence in its predictions. However, as the temperature decreases, gradients can vanish, leading to difficulties in training stability and convergence. This trade-off between stability and accuracy needs to be carefully managed to prevent degradation in model performance.

Moreover, the model's adaptability to different sequence ordering scenarios remains a challenge. The current approach may struggle with variations in the order of entity and relation sequences, affecting its ability to generalize across diverse data structures. Consistency in sequence ordering during training and inference is crucial, but the model needs to be more robust to different ordering to handle various real-world scenarios effectively.

In addition to these limitations, in real-life scenarios, limitations include the computational expense of beam search. While enhancing the quality of generated sequences, requires significant computational resources. This high computational cost makes it challenging to apply the model to large-scale datasets or real-time applications. The computational overhead can hinder the practical deployment of the model in resource-constrained environments. Techniques like adaptive beam search, where the beam size is dynamically adjusted based on the complexity of the task, could help balance performance and computational efficiency.

6.3.3 Future Directions

This work contributes to the broader goal of improving the accuracy and robustness of IE systems, laying a foundation for further advancements in the field of information extraction, and offering a range of practical applications. The model can be directly applied to automated text analysis tasks, such as information extraction from news articles, social media posts, and academic papers. Our approach can also be used to build and update knowledge graphs by accurately extracting entities, relations, and events from text.

Techniques for softening decoding strategies may benefit other NLP tasks such as translation, sentiment analysis, and summarization, where exposure bias and sequence alignment issues similarly affect performance. By addressing these common challenges, our methods can enhance the accuracy and reliability of various NLP applications.

Bibliography

1998. *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.*

Sherief Abdallah, Khaled F. Shaalan, and Muhammad Shoaib. 2012. Integrating rule-based system with classification for arabic named entity recognition. In *Conference on Intelligent Text Processing and Computational Linguistics*.

Ahmed Abdul-Hamid and Kareem Darwish. 2010. Simplified feature set for Arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, pages 110–115, Uppsala, Sweden. Association for Computational Linguistics.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1723–1729, Copenhagen, Denmark. Association for Computational Linguistics.

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12462–12470.

Muhammad Saleh Al-Qurishi and Riad Souissi. 2021. Arabic named entity recognition using transformer-based-CRF model. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 262–271, Trento, Italy. Association for Computational Linguistics.

Mohammad AL-Smadi and Omar Qawasmeh. 2016. Knowledge-based approach for event extraction from arabic tweets. *International Journal of Advanced Computer Science and Applications*, 7(6).

Mohammed G.H. Al Zamil and Qasem Al-Radaideh. 2014. Automatic extraction of ontological relations from arabic text. *Journal of King Saud University - Computer and Information Sciences*, 26(4):462–472. Special Issue on Arabic NLP.

Mohammed N. A. Ali, Guanzheng Tan, and Aamir Hussain. 2018. Bidirectional Recurrent Neural Network Approach for Arabic Named Entity Recognition. *Future Internet*, 10(12):1–12.

Nasser Alsaedi and Pete Burnap. 2015. Arabic event detection in social media. In *Computational Linguistics and Intelligent Text Processing*, pages 384–401, Cham. Springer International Publishing.

Sa'a D. A. Alzboun, Saia Khaled Tawalbeh, Mohammad Al-Smadi, and Yaser Jararweh. 2018. Using bidirectional long short-term memory and conditional random fields for labeling arabic named entities: A comparative study. *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 135–140.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Dhananjay Ashok and Zachary C. Lipton. 2023. Promptner: Prompting for named entity recognition.

Razieh Baradaran and Behrouz Minaei-Bidgoli. 2015. Event extraction from classical arabic texts. *The International Arab Journal of Information Technology (IAJIT).*, 12(5):494–502.

M Saiful Bari, Shafiq Joty, and Prathyusha Jwalapuram. 2020. Zero-resource cross-lingual named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7415–7423.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Abdelmajid Ben Hamadou, Odile Piton, and Héla Fehri. 2010. Multilingual Extraction of functional relations between Arabic Named Entities using NooJ platform. In *Nooj 2010 International Conference and Workshop*, pages 192–202, Komotini, Greece. Le Département de Philologie Grecque de l'Université Democritus de Thrace, le Laboratoire de Sémio-Linguistique et Didactique (LASELDI) de l'Université de Franche-Comté et la Maison des Sciences de l'Homme et de l'Environnement Ledoux. 10 pages.

Yassine Benajiba, Mona Diab, and Paolo Rosso. 2008a. Arabic named entity recognition using optimized feature sets. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 284–293, Honolulu, Hawaii. Association for Computational Linguistics.

Yassine Benajiba, Mona Diab, and Paolo Rosso. 2009. Using language independent and language specific features to enhance arabic named entity recognition. *The International Arab Journal of Information Technology (IAJIT).*, 6:463–471.

Yassine Benajiba, Mona T. Diab, and P. Rosso. 2008b. Arabic named entity recognition: An svm-based approach.

Yassine Benajiba, Mona T. Diab, and Paolo Rosso. 2004. Named entity recognition and classification for text in arabic. In *International Conference on Intelligent and Adaptive Systems and Software Engineering*.

Yassine Benajiba and Paolo Rosso. 2007. Anersys 2.0: Conquering the ner task for the arabic language by combining the maximum entropy with pos-tag information. In *Indian International Conference on Artificial Intelligence*.

Yassine Benajiba and Paolo Rosso. 2008. Arabic named entity recognition using conditional random fields. *Proceedinfs of Workshop on HLT & NLP within the Arabic World, LREC*, 8:143–153.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yassine Benajiba, Imed Zitouni, Mona Diab, and Paolo Rosso. 2010. Arabic named entity recognition: using features extracted from noisy data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, page 281–285, USA. Association for Computational Linguistics.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 148–151.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1171–1179, Cambridge, MA, USA. MIT Press.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Casimir Borkowski and Thomas J. Watson. 1967. An experimental system for automatic recognition of personal titles and personal names in newspaper texts. In *COLING 1967 Volume 1: Conference Internationale Sur Le Traitement Automatique Des Langues*.

Tim Buckwalter. *Arabic Morphological Analyzer Version 2.0 LDC2004L02*. Linguistic Data Consortium, 2004.

R Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias1. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Citeseer.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Dierong Chen, Juanjuan Cai, and Chuanzhen Li. 2021a. A joint entity-relation extraction method with sparse parameter sharing architecture. In *2021 7th International Conference on Computer and Communications (ICCC)*, pages 1705–1709.

Weile Chen, Huiqiang Jiang, Qianhui Wu, Börje Karlsson, and Yi Guan. 2021b. AdvPicker: Effectively Leveraging Unlabeled Data via Adversarial Discriminator for Cross-Lingual NER. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 743–753, Online. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Hai Leong Chieu and Hwee Tou Ng. 2002. Named entity recognition: A maximum entropy approach using global information. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Nancy Chinchor. 2001. Muc-7 dataset. Web Download. Philadelphia: Linguistic Data Consortium.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 423–429, Barcelona, Spain.

Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. 2008. *Supervised Learning*, pages 21–49. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5308–5314, Florence, Italy. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Cícero dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop*, pages 25–33, Beijing, China. Association for Computational Linguistics.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China. Association for Computational Linguistics.

Javid Ebrahimi and Dejing Dou. 2015. Chain based RNN for relation classification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1244–1249, Denver, Colorado. Association for Computational Linguistics.

Maud Ehrmann, Marco Turchi, and Ralf Steinberger. 2011. Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 118–124, Hissar, Bulgaria. Association for Computational Linguistics.

Niama El Khbir, Nadi Tomeh, and Thierry Charnois. 2022. ArabIE: Joint entity, relation and event extraction for Arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 331–345, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ali Elsebai, Farid Meziane, Fatma Zohra Belkredim, et al. 2009. A rule based persons names arabic extraction system. *Communications of the IBIMA*, 11(6):53–59.

MA Falih and N Omar. 2015. A comparative study on arabic grammatical relation extraction based on machine learning classification. *MiddleEast Journal of Scientific Research*, 23:1222–1227.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2022. LasUIE: Unifying information extraction with latent adaptive structure-aware generative language model. In *Advances in Neural Information Processing Systems*.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 168–171.

G.D. Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Karl Pearson F.R.S. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Ruiji Fu, Bing Qin, and Ting Liu. 2011. Generating Chinese named entity data from a parallel corpus. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 264–272, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Ignazio Gallo, Elisabetta Binaghi, Moreno Carullo, and Nicola Lamberti. 2008. Named entity recognition by neural sliding window. pages 567–573.

Abbas Ghaddar and Phillippe Langlais. 2017. WiNER: A Wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zoubin Ghahramani. 2004. *Unsupervised Learning*, pages 72–112. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2017. Differentiable scheduled sampling for credit assignment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 366–371, Vancouver, Canada. Association for Computational Linguistics.

Kartik Goyal, Graham Neubig, Chris Dyer, and Taylor Berg-Kirkpatrick. 2018. A continuous relaxation of beam search for end-to-end training of neural sequence models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Nizar Y. Habash. 2010. *Introduction to Arabic natural language processing*, 1 edition, volume 3 of *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool Publishers.

Rema Muftah Hamad and Ahmed Mohamed Abushaala. 2023. Medical named entity recognition in arabic text using svm. In *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, pages 200–205.

Fouzi Harrag and Selmene Gueliani. 2020. Event extraction based on deep learning in food hazard arabic texts.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

Chadi Helwe, Ghassan Dib, Mohsen Shamas, and Shady Elbassuoni. 2020. A semi-supervised BERT approach for Arabic named entity recognition. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 49–57, Barcelona, Spain (Online). Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenhao Huang, Jiaqing Liang, Zhixu Li, Yanghua Xiao, and Chuanjun Ji. 2023. Adaptive ordered information extraction with deep reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13664–13678, Toronto, Canada. Association for Computational Linguistics.

Mohammad Hudhud, Hamed Abdelhaq, and Fadi Mohsen. 2021. Arabianer: A system to extract named entities from arabic content. In *International Conference on Agents and Artificial Intelligence*.

K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. 1998. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Alankar Jain, Bhargavi Paranjape, and Zachary C. Lipton. 2019a. Entity projection via machine translation for cross-lingual NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1083–1092, Hong Kong, China. Association for Computational Linguistics.

Alankar Jain, Bhargavi Paranjape, and Zachary C. Lipton. 2019b. Entity projection via machine translation for cross-lingual NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1083–1092, Hong Kong, China. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested Arabic named entity corpus and recognition using BERT. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3626–3636, Marseille, France. European Language Resources Association.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain. Association for Computational Linguistics.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl$_1$) $: i180 - -i182$.

Yongil Kim, Yerin Hwang, Joongbo Shin, Hyunkyung Bae, and Kyomin Jung. 2023. Injecting comparison skills in task-oriented dialogue systems for database search results disambiguation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4047–4065, Toronto, Canada. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Rim Koulali and Abdelouafi Meziane. 2012. A contribution to arabic named entity recognition. In *2012 Tenth International Conference on ICT and Knowledge Engineering*, pages 46–52.

Onur Kuru, Ozan Arkan Can, and Deniz Yuret. 2016. CharNER: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 911–921, Osaka, Japan. The COLING 2016 Organizing Committee.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

Changki Lee. 2017. Lstm-crf models for named entity recognition. *IEICE Transactions on Information and Systems*, E100.D:882–887.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18.

Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. 2005. Svm based learning system for information extraction. In *Deterministic and Statistical Methods in Machine Learning*, pages 319–339, Berlin, Heidelberg. Springer Berlin Heidelberg.

Nut Limsopatham and Nigel Collier. 2016. Bidirectional LSTM for named entity recognition in Twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145–152, Osaka, Japan. The COLING 2016 Organizing Committee.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics*.

Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. Autoregressive structured prediction with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 993–1005, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

John Maloney and Michael Niv. 1998. TAGARAB: A fast, accurate Arabic name recognizer using high-precision morphological analysis. In *Computational Approaches to Semitic Languages*.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188–191.

Mohamed A. Meselhi, Hitham M. Abo Bakr, Ibrahim Ziedan, and Khaled Shaalan. 2014. A novel hybrid approach to arabic named entity recognition. In *Machine Translation*, pages 93–103, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Reham Mohamed, Nagwa M. El-Makky, and Khaled Nagi. 2015. Arabrelat: Arabic relation extraction using distant supervision. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, IC3K 2015, page 410–417, Setubal, PRT. SCITEPRESS - Science and Technology Publications, Lda.

Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A. Smith. 2012. Recall-oriented learning of named entities in Arabic Wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173, Avignon, France. Association for Computational Linguistics.

Hanane Nour Moussa and Asmaa Mourhir. 2023. Darnercorp: An annotated named entity recognition dataset in the moroccan dialect. *Data in Brief*, 48:109234.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Robert Munro and Christopher D. Manning. 2012. Accurate unsupervised joint named-entity extraction from unaligned parallel text. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 21–29, Jeju, Korea. Association for Computational Linguistics.

Nazlia Naji. 2012. Arabic named entity recognition using artificial neural network. *Journal of Computer Science*, 8:1285–1293.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021a. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022a. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022b. Learning cross-task dependencies for joint extraction of entities, events, event arguments, and relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9349–9360, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Minh Van Nguyen, Tuan Ngo Nguyen, Bonan Min, and Thien Huu Nguyen. 2021b. Crosslingual transfer learning for relation and event extraction via word category and class alignments. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5414–5426, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Mai Oudah and Khaled Shaalan. 2012. A pipeline Arabic named entity recognition using a hybrid approach. In *Proceedings of COLING 2012*, pages 2159–2176, Mumbai, India. The COLING 2012 Organizing Committee.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Nita Patil, Ajay Patil, and BV Pawar. 2020. Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Longhua Qian, Haotian Hui, Ya'nan Hu, Guodong Zhou, and Qiaoming Zhu. 2014. Bilingual active learning for relation classification via pseudo parallel corpora. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Juan Enrique Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

L.F. Rau. 1991. Extracting company names from text. In *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, volume i, pages 29–32.

Douglas A. Reynolds. 2009. Gaussian mixture models. In *Encyclopedia of Biometrics*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. *Learning internal representations by error propagation*, page 318–362. MIT Press, Cambridge, MA, USA.

Evan Sandhaus. 2008. The new york times annotated corpus ldc2008t19.

Injy Sarhan, Yasser El-Sonbaty, and Mohamad Abou El-Nasr. 2016. Semi-supervised pattern based algorithm for arabic relation extraction. *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 177–183.

Khaled Shaalan and Hafsa Raza. 2007. Person name entity recognition for Arabic. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 17–24, Prague, Czech Republic. Association for Computational Linguistics.

Khaled F. Shaalan and Hafsa Raza. 2008. Arabic named entity recognition from diverse text types. In *GoTAL*.

Khaled F. Shaalan and Hafsa Raza. 2009. Nera: Named entity recognition for arabic. *J. Assoc. Inf. Sci. Technol.*, 60:1652–1663.

Aditya Sharma, Zarana Parekh, and Partha Talukdar. 2017. Speeding up reinforcement learning-based information extraction training using asynchronous methods. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2658–2663, Copenhagen, Denmark. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 313–325, Hong Kong, China. Association for Computational Linguistics.

Nasrin Taghizadeh, Heshaam Faili, and Jalal Maleki. 2018. Cross-language learning for arabic relation extraction. *Procedia Computer Science*, 142:190–197.

Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.

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

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hayssam Traboulsi. 2009. Arabic named entity extraction: A local grammar-based approach. In *2009 International Multiconference on Computer Science and Information Technology*, pages 139–143.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2020. Revisiting unsupervised relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7498–7505, Online. Association for Computational Linguistics.

Tzong-Han Tsai, Chia-Wei Wu, and Wen-Lian Hsu. 2005. Using maximum entropy to extract biomedical named entities without dictionaries. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

C. Walker and Linguistic Data Consortium. 2005. *ACE 2005 Multilingual Training Corpus*. LDC corpora. Linguistic Data Consortium.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Qianhui Wu, Zijia Lin, Börje F. Karlsson, Biqing Huang, and Jian-Guang Lou. 2020a. Unitrans : Unifying model transfer and data transfer for cross-lingual named entity recognition with unlabeled data. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3926–3932. International Joint Conferences on Artificial Intelligence Organization. Main track.

Qianhui Wu, Zijia Lin, Guoxin Wang, Hui Chen, Börje F. Karlsson, Biqing Huang, and Chin-Yew Lin. 2020b. Enhanced meta-learning for cross-lingual named entity recognition with minimal resources. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9274–9281.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783, Copenhagen, Denmark. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3533–3546, Online. Association for Computational Linguistics.

Wang Xu, Kehai Chen, Lili Mou, and Tiejun Zhao. 2022. Document-level relation extraction with sentences importance estimation and focusing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2920–2929, Seattle, United States. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pages 5708–5717. PMLR.

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2019. Joint extraction of entities and relations based on a novel decomposition strategy.

Chenhan Yuan and Hoda Eldardiry. 2021. Unsupervised relation extraction: A variational autoencoder approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1929–1938, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Urchade Zaratiana, Niama Elkhbir, Pierre Holat, Nadi Tomeh, and Thierry Charnois. 2022a. Global span selection for named entity recognition. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 11–17, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2022b. Named entity recognition as structured span prediction. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 1–10, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. Kernel methods for relation extraction. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 71–78. Association for Computational Linguistics.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint Entity and Event Extraction with Generative Adversarial Imitation Learning. *Data Intelligence*, 1(2):99–120.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2021. A unified multi-task learning framework for joint extraction of entities and relations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14524–14531.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346, Hong Kong, China. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, Michigan. Association for Computational Linguistics.

Ayah Zirikly and Mona Diab. 2014. Named entity recognition system for dialectal Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 78–86, Doha, Qatar. Association for Computational Linguistics.

Ayah Zirikly and Mona Diab. 2015. Named entity recognition for Arabic social media. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 176–185, Denver, Colorado. Association for Computational Linguistics.

Arkaitz Zubiaga. 2018. Twitter event datasets (2012-2016).