Multilingual Information Extraction: Challenges and Solutions

Minh Van Nguyen
Department of Computer and Information Science, University of Oregon,
Eugene, OR 97403, USA
minhnv@cs.uoregon.edu

Abstract

Despite the existence of approximately 7000 languages, research in Natural Language Processing has predominantly focused on a select few high-resource languages, which does not serve the global linguistic diversity adequately. Multilingual Information Extraction aims to improve information access and communication across various languages, has therefore emerged as a vital research area. This field entails several key tasks, namely Event Trigger Detection, Event Argument Extraction, Entity Mention Recognition, and Relation Extraction, each contributing to the extraction of structured information from unstructured text. This work explores three primary research directions in Multilingual IE: (1) enhancing Multilingual IE upstream models, (2) developing language-agnostic downstream models, and (3) advancing cross-lingual transfer learning methods for situations with scarce training data. These directions are examined in detail, highlighting the recent advancements, enduring challenges, and future prospects, contributing to the overarching goal of democratizing communication and information access in the linguistic landscape of our world.

1 Introduction

In our increasingly interconnected world, the demand for efficient communication and understanding across languages has never been greater. An essential cornerstone of this global dialogue lies in the capacity to parse, process, and interpret information in its myriad forms and linguistic structures. It is here that the field of Multilingual Information Extraction (Multilingual IE) rises to prominence. Nestled within the broader ambit of Natural Language Processing (NLP), Multilingual IE is charged with the critical task of distilling structured information from unstructured text in a diverse array of languages (Pouran Ben Veyseh et al., 2022b,a; Lai et al., 2022a,b).

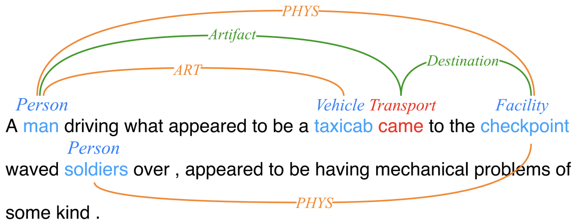

Figure 1: An example with annotations of the four IE tasks (Nguyen et al., 2021a).

Approximately 7000 languages punctuate our global linguistic landscape, each rich with its unique syntax, semantic nuances, and linguistic idiosyncrasies. Despite this diversity, the lion's share of NLP research has primarily been focused on a select few high-resource languages, with English often at the forefront (Adelani et al., 2021). This disproportionate focus is increasingly being recognized as a bottleneck that curtails the comprehensive service of global linguistic diversity. Bridging this gap, Multilingual NLP, and more specifically, Multilingual IE, has emerged as a vital area of research, striving to democratize information access and catalyze communication for individuals spanning the gamut of linguistic diversity (Pouran Ben Veyseh et al., 2022b,a; Lai et al., 2022a,b).

At the heart of Multilingual IE lie several interconnected sub-tasks, namely Event Trigger Detection (ETD), Event Argument Extraction (EAE), Entity Mention Recognition (EMR), and Relation Extraction (RE) (see Figure 1). Each of these tasks carves out its unique niche within the overarching task of information extraction (Nguyen and Grishman, 2016; Nguyen et al., 2016; Nguyen and Grishman, 2018). For example, EMR is focused on the identification of entities within a text, while RE is involved in identifying relationships between these recognized entities. ETD and EAE, on the other hand, concentrate on detecting events and their related arguments embedded within the textual discourse. The heterogeneity of these tasks highlights the intricate labyrinth of language processing, revealing the multi-pronged challenges faced in converting raw, unstructured text into structured, usable information.

Notwithstanding the inherent challenges, the field of Multilingual IE has witnessed significant advancements. To elaborate on these strides, this paper will dissect three primary research trajectories within Multilingual IE:

- **Multilingual IE Upstream Models**: These aim to refine the quality of upstream linguistic features to bolster the performance of downstream IE models.

- **Language-agnostic Downstream Models for IE**: These seek to design IE models with language-agnostic architectures, thereby enabling their deployment across a multitude of languages without necessitating model-specific tweaks.

- **Cross-lingual Transfer Learning for IE**: This direction tackles scenarios where training data for IE tasks in a target language is scarce or nonexistent. It seeks to develop learning methods such that a model trained on a source language can be seamlessly ported to a target language.

Each of these research directions will be examined through the lens of a myriad of seminal and contemporary studies, elucidating their methodologies, outcomes, and implications. This rigorous dissection will not only spotlight the strides achieved in the field but will also bring into sharp relief the challenges that endure and the prospects that lie ahead.

The examination of multilingual IE upstream models will delve into the enhancement of fundamental linguistic features such as sentence boundaries, word tags, and dependency trees, that form the backbone of downstream IE models (Pouran Ben Veyseh et al., 2019, 2020). The limitations of current NLP toolkits, such as speed, performance, and limited language support, are being addressed by leveraging transformer-based language models and adapter methods (Pfeiffer et al., 2020; Nguyen et al., 2021b). Despite these advancements, the issue of model size and consequent memory use remains a challenge, inspiring research into model compression techniques (Mysore Sathyendra et al., 2020; Park et al., 2021).

In exploring the language-agnostic downstream models for IE, we shift the focus from the foundational linguistic features to the architecture of the IE models themselves. Historically, pipelined approaches were used, where models for one task consumed the output of models performing other tasks (Zhou et al., 2005; Nguyen and Grishman, 2015; Lai et al., 2020). However, error propagation from one task to another posed a significant challenge. A paradigm shift towards joint models, known as Joint Information Extraction (JointIE), has since emerged, where ETD, EMR, EAE, and RE are all performed in a single model to mitigate error propagation (Luan et al., 2019; Lin et al., 2020; Nguyen et al., 2021a, 2022a,b) and leverage the interdependencies between the tasks. Nonetheless, further research is required to fully leverage the language differences and similarities for improved multilingual training and model generalization.

The final research direction, cross-lingual transfer learning for IE, considers scenarios with limited or nonexistent training data in a target language. Strategies such as multilingual word embeddings (Chen and Cardie, 2018; Heyman et al., 2019) or multilingual pre-trained language models (Devlin et al., 2019; Conneau et al., 2019) have been used to learn crosslingual representation vectors for IE. Yet, these approaches have suffered the issue of monolingual bias, stemming from models trained solely on source language data, leading to suboptimal crosslingual performance. More recent strategies like language adversarial training (Chen et al., 2019; Huang et al., 2019; Lange et al., 2020) have aimed to address this issue, but further research is necessary for more robust crosslingual performance in IE.

In essence, the breadth and depth of Multilingual IE, while presenting significant challenges, also beckon exciting opportunities for research. The relentless quest for enhanced performance, reduced error propagation, improved model efficiency, and superior crosslingual capabilities drive the field forward. As we peel back the layers of each research direction, we shine a light on the potential for significant advancements in how we extract, process, and interpret multilingual information, transforming the way we communicate and access information in our increasingly interconnected world.

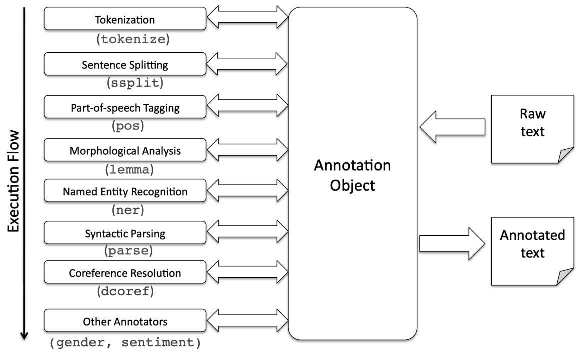

Figure 2: Overview of Stanford CoreNLP's architecture (Manning et al., 2014).

2 Multilingual IE Upstream Models

2.1 The Stanford CoreNLP Natural Language Processing Toolkit

Stanford CoreNLP (Manning et al., 2014) was originally developed for internal purposes. The task at hand involved unifying a variety of natural language analysis components, each having distinct ad hoc APIs, with specific glue code. The first iteration of the annotation pipeline was created in 2006 to enhance this amalgamation. A unified interface was established for an Annotator to integrate analytical data into the text. This was accomplished by the Annotator adding additional details to an Annotation object, which was stored as a typesafe heterogeneous map, inspired by Bloch's data type concepts (Bloch, 2008). This simple architecture has demonstrated considerable success and continues to be the system's foundation, as depicted in figure 2. The motivations included:

- Acquiring linguistic annotations for a text quickly and without difficulty.
- Concealing differences between components behind a shared API.
- Maintaining a minimal conceptual footprint for ease of understanding.
- Offering a lightweight framework utilizing plain Java objects, rather than complex elements such as XML or UIMA's Common Analysis System (CAS) objects.

In 2009, the system was expanded for accessibility by a wider user base as part of a multi-site grant project. This included providing a command-line interface and outputting an Annotation in various formats, like XML. Continued improvements led to the public release of the system as free open source software in 2010.

Stanford CoreNLP, from an architectural standpoint, does not aim for exhaustive functionality. It primarily offers a straightforward pipeline architecture with a Java API. It does not endeavor to support multi-machine scaling but does allow for multi-threaded processing on a single device. While the system meets the needs of a significant user base, its simplicity eases the learning curve. In comparison to larger frameworks like UIMA (Ferrucci and Lally, 2004) or GATE (Cunningham et al., 2002), users can start using Stanford CoreNLP with a basic understanding of Java. More complex scenarios like multi-machine scale-out can typically be handled by running the analysis pipeline within a distributed workflow system like Hadoop or Spark. Other systems aim to provide more features, like the UIUC Curator (Clarke et al., 2012), which includes inter-machine client-server communication and caching for natural language analyses. However, this added functionality complicates installation and understanding of the system. Furthermore, an organization might already be committed to a different scale-out solution not provided by the natural language analysis toolkit, for instance, Kryo or Google's protobuf for binary serialization as opposed to Apache Thrift used by the Curator. Users are better suited by a compact and self-contained natural language analysis system without unnecessary complexities.

On the contrary, users greatly benefit from a collection of stable, high-quality linguistic analysis components that are easily accessible for common scenarios. Developers designing larger systems may have made broad design choices like scale-out handling, but they might not be NLP experts, therefore requiring NLP components that are straightforward. This advantage sets Stanford CoreNLP and GATE apart from the bare-bones package of an Apache UIMA download. Solutions provided by well-integrated component packages for UIMA, such as ClearTK (Bethard et al., 2014), DKPro Core (Gurevych et al., 2007), and JCoRe (Hahn et al., 2016), continue to be more complex and heavier for users compared to the pipeline discussed. The success factors of Hibernate, as outlined by Patricio (2009), are echoed in these characteristics. These include: doing one thing exceptionally, avoiding over-design, and ensuring that the system is operational in ten minutes or less. Other factors that Patricio emphasizes, such as shunning standardism, good documentation, and developer responsiveness, are also mirrored in the design and success of Stanford CoreNLP. While there are numerous factors that influence the adoption of a project, it's believed that some of these characteristics explain why Stanford CoreNLP is one of the more popular NLP toolkits. Despite not being perfect, Stanford CoreNLP has gained popularity due to clear open source licensing, sufficient documentation, and efforts to answer user queries, setting it apart from much of the academic software.

2.2 UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing

UDPipe (Straka et al., 2016) is a comprehensive tool developed under the Universal Dependencies project (de Marneffe et al., 2014), aimed at facilitating cross-linguistic consistent treebank annotation for a multitude of languages. The scheme for annotation is founded on the universal Stanford dependencies (Manning et al., 2014), Google's universal part-of-speech tags (Nivre et al., 2016), and the Interset interlingua for morphosyntactic features (Zeman, 2008).

The design and creation of UDPipe is driven by a desire to produce an easy-to-use tool that would enable the processing of raw text into CoNLL-U-formatted tagged and/or parsed dependency trees. The tool's design goals include creating state-of-the-art tools for tokenization, morphological analysis, part-of-speech tagging and dependency parsing. Furthermore, the tool is aimed to be easy to train with custom data, provided in CoNLL-U format. The tool is built on efficient programming design, optimizing both RAM and disk usage.

UDPipe is a C++ tool available under the Mozilla Public License (MPL) 2.0 license (code) and CC BY-NC-SA 4.0 license (models). The single-model tool (per language) is designed for simplicity, eschewing feature engineering, external morphological dictionaries, and language-specific knowledge. The project seeks to provide trained models for as many UD treebanks as possible.

Tokenization is a critical process and is often considered a simple task for many languages, especially those that use separators between words. However, specific rules are often applied to partition unseparated words or replace contractions in many languages. To this end, UDPipe includes a trainable tokenizer based on artificial neural networks.

While parsing the morphological fields in the CoNLL-U format, UDPipe is capable of filling in universal part-of-speech tags, lists of morphological features, language-specific part-of-speech tags, and lemma or stems, depending on the availability of these in the training data. It employs MorphoDiTa (Straková et al., 2014) for POS tagging and lemmatization, which utilizes a supervised, rich feature averaged perceptron, employing dynamic programming at runtime (Viterbi decoder).

To bypass the need for a language-specific code or additional language resources, UDPipe develops a morphological "guesser." For every suffix of fixed length, the tool identifies the most frequent analyses according to the training data, creating a morphological dictionary from the UD data.

UDPipe internally uses two models, one for disambiguating all available morphological fields (a POS tagger), and the other one for lemmatization (a lemmatizer). This combination of two taggers improves overall accuracy. The POS tagger disambiguates all available morphological fields and can jointly disambiguate a lemma as well, enhancing tagging accuracy.

UDPipe employs the Parsito parser, a transition-based, non-projective dependency parser that uses a neural network classifier for prediction and requires no feature engineering. It's capable of parsing both projective and non-projective sentences. The parser has an exceptional parsing speed, compact models, and delivers high accuracy.

The tool can be obtained from the UDPipe homepage or directly using the permanent ID. The training code is included in the release, and the entire pipeline is easily trainable using training data in CoNLL-U format. All the trained models of the whole pipeline are stored in a single file, with options to train only a selected part of the complete pipeline. The tool offers high throughput, with model sizes on the order of megabytes.

Moreover, UDPipe is available as a library with many language bindings – Java, Python, Perl, and C# are currently offered. Furthermore, UDPipe is available as a web service with a REST API, enabling users to access its functionalities from anywhere via the internet.

In the web service, a simple HTTP POST request is sent to the UDPipe server with raw text data and the chosen model for the language of the text. The server then processes the text and returns the output in the CoNLL-U format, which can be conveniently processed further in any text-processing pipeline.

As for further developments, UDPipe continually aims to improve its functionality. The team regularly trains models with each new release of the Universal Dependencies treebanks. In addition, they continually work on refining the parsing algorithm, improving speed and accuracy, and adding support for new languages as Universal Dependencies expands.

It's also worth noting that UDPipe has extensive documentation that covers every aspect of the tool, from installation to usage and training custom models. This makes it easy for new users to get started and for experienced users to customize and optimize the tool for their specific needs.

In terms of applicability, UDPipe is a robust tool for NLP professionals and researchers who work with multiple languages. It can be used in a wide array of tasks like sentiment analysis, machine translation, information extraction, and more. Given its design to handle different languages consistently, it's a particularly useful tool for projects that involve cross-linguistic studies or applications.

Overall, UDPipe is a comprehensive tool that provides end-to-end NLP processing, designed with the vision of promoting Universal Dependencies as a standard for cross-linguistic consistent treebank annotation. By providing an accessible, efficient, and powerful tool, it aims to facilitate and enhance NLP work across a multitude of languages.

2.3 Stanza: A Python Natural Language Processing Toolkit for Many Human Languages

Stanza (Qi et al., 2020) is comprised of two distinct components: a neural multilingual natural language processing (NLP) pipeline and a Python client interface to the Java Stanford CoreNLP software. An illustration of Stanza pipeline is provided in Figure 3.

Stanza's neural pipeline contains models that span from tokenizing raw text to performing syntactic analysis on entire sentences. These components are built with the aim of processing a variety of human languages. High-level design choices encapsulate common phenomena across multiple languages, and data-driven models learn the distinctions among these languages from the gathered data. The implementation of Stanza components is notably modular, and it tends to reuse basic model architectures for efficiency and compactness.

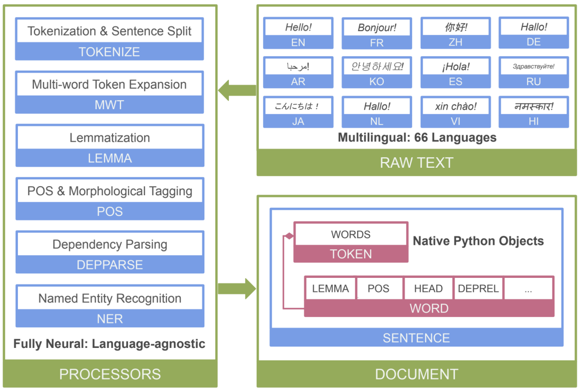

Figure 3: Stanza's system design (Qi et al., 2020).

When presented with raw text, Stanza tokenizes it and groups tokens into sentences. It integrates both tokenization and sentence segmentation from raw text into a single module, treated as a tagging problem over character sequences. This module predicts if a given character signals the end of a token, sentence, or multi-word token (MWT). MWTs are predicted alongside tokenization because this task is context-sensitive in certain languages.

Once MWTs are identified, they are expanded into the underlying syntactic words to set the groundwork for downstream processing. This expansion is accomplished with a combination of a frequency lexicon and a neural sequence-to-sequence model (Edunov et al., 2018). This ensures that commonly observed expansions in the training set are robustly expanded while maintaining flexibility to model unseen words statistically.

For each word in a sentence, Stanza assigns it a part-of-speech (POS), and analyzes its universal morphological features. This prediction of POS and universal morphological features is carried out by adopting a bidirectional long short-term memory network. To ensure consistency among universal POS, treebank-specific POS, and universal morphological features, the biaffine scoring mechanism is utilized.

In addition to these functionalities, Stanza lemmatizes each word in a sentence to retrieve its canonical form. Like the multi-word token expander, Stanza's lemmatizer is implemented as an ensemble of a dictionary-based lemmatizer and a

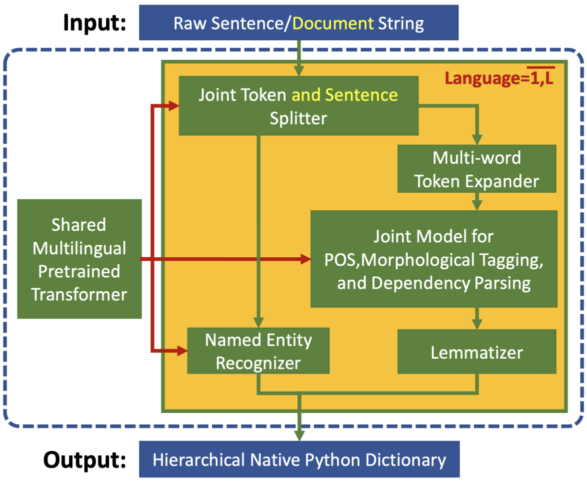

Figure 4: Trankit's system design (Nguyen et al., 2021b).

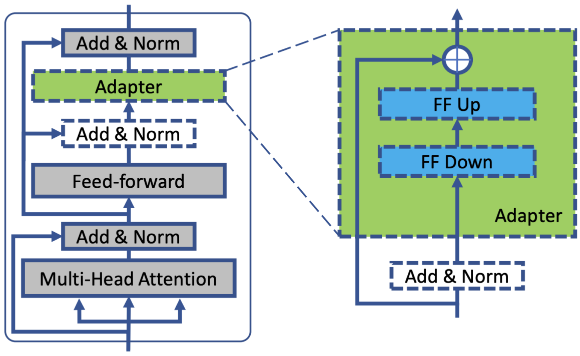

Figure 5: Adapter networks used in Trankit (Nguyen et al., 2021b).

neural sequence-to-sequence lemmatizer. An extra classifier is constructed on the encoder output of the sequence-to-sequence model, to predict shortcuts such as lowercasing and identity copy for robustness on lengthy input sequences such as URLs.

Stanza parses each sentence for its syntactic structure, with each word in the sentence being assigned a syntactic head that is either another word in the sentence, or in the case of the root word, an artificial root symbol. This parsing is conducted through a Bi-LSTM-based deep biaffine neural dependency parser (Dozat and Manning, 2017), with the addition of two linguistically motivated features: one predicting the linearization order of two words in a given language, and the other predicting the typical distance in linear order between them.

Stanza also recognizes named entities in each input sentence, like person names and organizations. For this task, the contextualized string representation-based sequence tagger is employed. This process involves training a forward and a backward character-level LSTM language model, and at tagging time, the representations at the end of each word position from both language models with word embeddings are concatenated, and fed into a standard one-layer Bi-LSTM sequence tagger with a conditional random field-based decoder (Wallach, 2004).

2.4 Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing

Trankit (Nguyen et al., 2021b) represents a solution to various challenges associated with multilingual Natural Language Processing (NLP) systems. Various efforts have been made to establish these systems in an attempt to break down language barriers. Much of this work has concentrated on downstream NLP tasks that heavily rely on upstream linguistic features. This extends from basic information like token and sentence boundaries in raw text to more complex structures such as part-of-speech tags, morphological features, and dependency trees of sentences, which are referred to as fundamental NLP tasks.

By constructing efficient multilingual systems or pipelines for these fundamental upstream NLP tasks, a transformation of multilingual downstream systems is possible. Several NLP toolkits addressing multilingualism for fundamental NLP tasks have been created, including notable examples such as spaCy[1], UDify(Kondratyuk and Straka, 2019), Flair(Akbik et al., 2019), CoreNLP (Manning et al., 2014), UDPipe (Straka, 2018), and Stanza (Qi et al., 2020). These toolkits, however, come with their own set of limitations. For example, while spaCy is built for speed, it compromises on performance. UDify and Flair are unable to process raw text as they rely on external tokenizers. CoreNLP does process raw text but does not deliver state-of-the-art performance. UDPipe and Stanza, more recent toolkits, utilize word embeddings to offer excellent performance for many languages, but their pipelines for different languages are trained separately, leading to high memory usage when multiple languages are needed. Moreover, none of these toolkits have explored the use of contextualized embeddings from pretrained transformer-based language models, which have shown promise in enhancing the performance of NLP tasks.

[1] https://spacy.io/

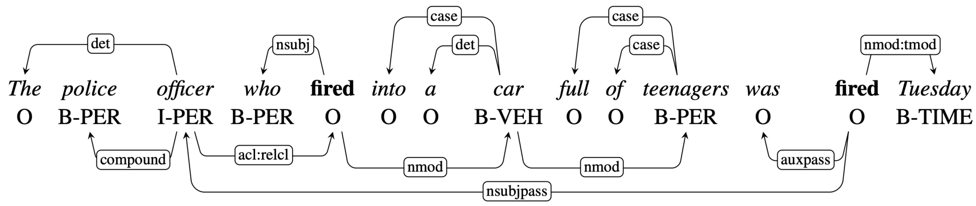

Figure 6: The annotation technique for entity mentions utilizes BIO notation. In this notation, the "B-X" label signifies the commencement of an entity mention of type "X", while "I-X" is designated for tokens that fall within (but do not initiate) the scope of an entity mention of type "X". Tokens that are not part of any entity mentions are marked with the "O" label. In the presented diagram, "PER" and "VEH" are used to represent PERSON and VEHICLE respectively. Furthermore, the universal dependency relations method is employed to construct a dependency parse tree for the example sentence (Nguyen and Grishman, 2018).

This is where Trankit comes in as a multilingual Transformer-based NLP Toolkit designed to surmount these limitations. Capable of processing raw text for fundamental NLP tasks, it supports 56 languages with 90 pre-trained pipelines based on the Universal Dependency v2.5 treebanks (Zeman et al., 2019). By leveraging the advanced multilingual pretrained transformer XLM-Roberta (Conneau et al., 2019), Trankit enhances the performance for sentence segmentation, part-of-speech tagging, morphological feature tagging, and dependency parsing while competing or even surpassing performance for tokenization, multi-word token expansion, and lemmatization across the 90 treebanks. It also matches or exceeds performance for named entity recognition on 11 public datasets (Mohit et al., 2012; Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003; Benikova et al., 2014; Weischedel et al., 2013; Nothman et al., 2013).

Uniquely, the token and sentence splitter in Trankit is based on wordpieces rather than characters to better exploit contextual information. This approach proves beneficial across many languages and is believed to be the first successful attempt to construct a wordpiece-based token and sentence splitter (Kudo and Richardson, 2018) that effectively functions for 56 languages.

The architecture of the Trankit pipeline features three innovative transformer-based components for token and sentence splitting, a joint model for POS tagging, morphological tagging, dependency parsing, and a named entity recognizer (see Figure 4). However, using a large pretrained transformer model like XLM-Roberta could pose memory issues as different transformer-based components need to be loaded into the memory for one or multiple languages to serve multilingual tasks. To tackle this, a novel plug-and-play mechanism with Adapters (Pfeiffer et al., 2020) is introduced. Adapters are small networks injected inside all layers of the pretrained transformer model. They are seen as an effective, lightweight alternative for the traditional fine-tuning of pretrained transformers (see Figure 5).

Within Trankit, a set of adapters and task-specific weights are created for each transformer-based component for each language, with only a single large multilingual pretrained transformer being shared across components and languages. This mechanism not only solves the memory problem but also significantly reduces the training time.

3 Language-agnostic Downstream Models for IE

3.1 Graph Convolutional Networks with Argument-Aware Pooling for Event Detection

Event Detection (ED) is a crucial aspect of natural language processing that recognizes specific instances of events, or event mentions, in text. These mentions typically exist within a single sentence, where an event trigger, usually a verb or nominalization, is associated with the event. The goal of ED is to identify these triggers and classify them into distinct categories. For instance, in a sentence like "The police officer who fired into a car full of teenagers was fired Tuesday", an ED system should discern that the first instance of "fired" is an Attack event while the second instance is an End-Position event. This task poses a challenge, as the same expression can signify different events based on context, and an event can be expressed in a variety of ways.

The most advanced method for ED involves deep learning models utilizing convolutional neural networks (CNNs) (Nguyen and Grishman, 2015). In basic implementation, CNNs execute temporal convolution operation on consecutive k-grams in sentences, producing latent structures valuable for ED. However, this consecutive convolution struggles to detect non-consecutive k-grams that might be crucial for identifying certain event triggers. For instance, in the example above, the non-consecutive 3-grams "officer was fired" should be identified to correctly assign the End-Position event to the second "fired". A non-consecutive CNN model (NCNN) (Nguyen and Grishman, 2016) was designed to counteract this by performing temporal convolution on all non-consecutive k-grams in sentences, becoming the state-of-the-art CNN model for ED.

Yet, by considering all non-consecutive k-grams, NCNNs might model unnecessary and noisy information that could potentially degrade the ED prediction performance. For example, the non-consecutive 3-gram "car was fired" in the given example could be misleading, suggesting an Attack event for the second "fired" instead of the accurate End-Position event. One way to avoid such confusion is by observing that "police officer" is directly related to the second "fired" in the example, while "a car" has no direct relation.

In (Nguyen and Grishman, 2018), a new approach is proposed to address this issue by performing convolution operation on syntactic dependency graphs of sentences (see Figure 6). These graphs represent sentences as directed trees with dependency arcs between related words. Performing convolution on these graphs helps focus on the most relevant words and avoid unrelated sequences. Experiments show that these syntactic connections provide efficient constraints for ED.

To achieve this, the paper utilizes Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) to form connections between layers of neural networks using graph structures. In GCNs, each node's convolution vector is calculated from the vectors of its immediate neighbors. This can be particularly useful for ED as the convolution vector of the current word can be used for prediction. However, the convolution vector may not encapsulate specific information about entity mentions that are distributed at various positions in the sentences. Such information is critical as it can provide models with more confidence to make predictions for ED.

To overcome this issue, this paper suggests using a pooling method over the graph-based convolution vectors of the current word and the entity mentions in the sentences. By aggregating the convolution vectors into a single vector representation for event type prediction, this method allows for explicit modeling of information from the entity mentions to improve ED performance. This proposed method is thoroughly evaluated with both manually annotated and automatically predicted entity mentions to demonstrate its effectiveness in the experiments.

3.2 Multi-View Consistency for Relation Extraction via Mutual Information and Structure Prediction

Relation Extraction is a key component of Information Extraction (IE), involving discerning semantic connections between entities mentioned in a text. This process is fundamental to numerous Natural Language Processing applications, like knowledge base population and question answering. As such, it has garnered significant interest from the NLP community, as evidenced by the wealth of research conducted on this topic in recent years.

Initially, the methodologies for this extraction were largely feature-based or kernel-based, relying heavily on feature engineering to create efficient models (Zelenko et al., 2002; Zhou et al., 2005). However, more recently, the focus has shifted towards deep learning models, which have improved performance considerably on relational extraction benchmark datasets. A pivotal technique introduced by deep learning has been the use of syntactic trees (like dependency trees and constituent trees) to structure computational graph networks (Zhang et al., 2018). The advantages of this approach include incorporating semantic/syntactic hierarchies into sentence representations and the ability to capture critical context words for relation extraction via dependency paths.

However, this method has limitations, primarily due to its reliance on high-quality external parsers to generate effective parse trees. This reliance presents three main issues: the availability of high-quality parsers is often limited to certain domains and languages, which restricts the usage of relation extraction models in specific situations. The tree structures provided by these parsers are not

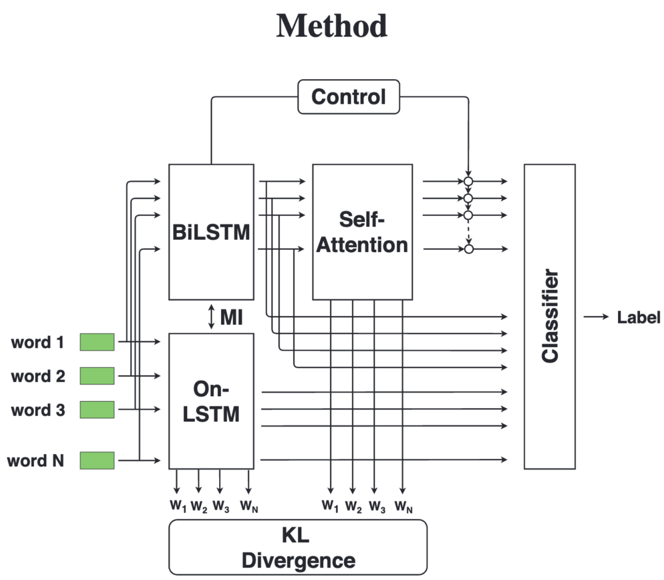

Figure 7: Proposed model for relation extraction by (Veyseh et al., 2020). The green vectors represent input word representations while the circles indicate the element- wise product.

always ideal for relation extraction, necessitating more task-specific structures. Lastly, the parsers' performance might decrease when applied to unfamiliar domains, which could detrimentally impact the relation extraction models that heavily rely on the quality of the tree structures.

To reduce this reliance on external parsers, this paper (Veyseh et al., 2020) proposes to learn implicit sentence structures along with predicting relationships between entities. This approach not only eliminates the need for external parsers but also enables the use of relation extraction models across different languages and domains. It offers task-specific sentence structures and has the potential to enhance the performance of relation extraction in new domains. The authors suggest learning sentence structures by implementing two different methods for generating dependencies/hierarchies between words in sentences. A measure of consistency between the structures and representations learned by these two methods is introduced to encourage congruity. The consistency, once enforced, is anticipated to expose effective task-specific structures and representations for relation extraction, an attribute unattainable with structures pre-determined by external parsers.

The two methods suggested for inducing sentence structures are Ordered Neurons Long-short Term Memory (ON-LSTM) (Shen et al., 2019) and self-attention (Vaswani et al., 2017), also known as Transformers. ON-LSTM, an advanced version of LSTM, assigns importance scores to each word in a sentence. These scores determine the word's closeness to the root of the tree structure, thereby implicitly forming a tree structure. In contrast, self-attention estimates the connection scores between every pair of words, forming a fully connected graph structure. The score for each pair reflects the influence one word has on understanding the semantics of the other word. To encourage structure consistency, the pairwise scores from self-attention are transformed into importance scores, and similarity between the two importance score sequences is promoted using the KL divergence in the loss function.

Although the authors consider the importance scores of every word in the input sentence for both ON-LSTM and self-attention in the baseline similarity promotion, only a subset of the words might be necessary for accurately recognizing relationships for entity mentions in relation extraction. Therefore, a filtering technique is proposed that predicts relevant context words and incorporates this information into the similarity promotion process to further enhance the structures for relation extraction.

However, a possible issue arises with the word representations produced by ON-LSTM and self-attention. They might become excessively constrained to achieve score similarity for structure consistency, potentially losing crucial semantic information for relation extraction. To mitigate this issue, the authors propose constraints to preserve important semantic information in the representation vectors produced by ON-LSTM and self-attention. They first use a bidirectional LSTM (BiLSTM) model to encode the semantic representations of words in its hidden vectors, then enrich the semantic content of the hidden vectors from ON-LSTM and self-attention via semantic consistency. Two mechanisms are considered to achieve this semantic consistency. One is inspired by the control mechanism that retains semantic content in the representation vectors of self-attention via the control vector computed from the BiLSTM vectors of the two entity mentions. The other mechanism, proposed in this work, leverages the mutual information between the high-dimension representation vectors from ON-LSTM and BiLSTM to ensure their semantic consistency.

3.3 Graph Transformer Networks with Syntactic and Semantic Structures for Event Argument Extraction

Event Extraction (EE) is a crucial facet of Information Extraction, with the aim to recognize events and their relevant arguments within a given text. In prior studies, EE is generally broken down into two primary tasks: Event Detection (ED) - the identification of event-triggering words, and Event Argument Extraction (EAE) - identifying event arguments and their respective roles based on given event triggers. Recently, significant attention has been directed towards the study of ED, applying deep learning techniques. Conversely, EAE has seen less exploration, even though it's vital for completing EE and benefits numerous downstream applications (Wang et al., 2019). The present study centers its attention on EAE to address this gap.

Modern leading methods for EAE engage deep learning models to compute an abstract representation vector for each word in input sentences. This computation is based on information gleaned from other context words. The representation vectors are then combined to execute EAE (Chen et al., 2015; Nguyen et al., 2016). The primary motivation behind the current work is to exploit different structures within input sentences, thereby improving the representation vectors for words in deep learning models designed for EAE. Here, a sentence structure (or view) refers to an importance score matrix. Each cell of this matrix quantifies the contribution of a context word towards the computation of the current word's representation vector for EAE. In particular, two types of sentence structures, syntactic and semantic structures, are considered.

For instance, let's consider the sentence:

"Iraqi Press constantly report interviews with Hussain Molem, the Hanif Bashir's son-in-law, while US officials confirmed all Bashir's family members were killed last week."

In this sentence, an EAE system should be able to recognize the entity mention "Hussain Molem" as the Victim of the Attack event triggered by "killed". As "Hussain Molem" and "killed" are situated far apart in the sentence and its dependency tree, EAE models may find it challenging to predict accurately. To overcome this, the models should first rely on the direct connections between "killed" and "all Bashir's family members" in the dependency tree, capturing the role of "all Bashir's family members" in the representation vectors for "killed". Following this, the models can exploit the close semantic similarity between "all Bashir's family members" and "the Hanif Bashir's son-in-law", further connecting "the Hanif Bashir's son-in-law" to "killed". Ultimately, the direct apposition relation between "the Hanif Bashir's son-in-law" and "Hussain Molem" can be used to link "Hussain Molem" with "killed" to obtain the necessary representations to predict arguments for "Hussain Molem". This example suggests both syntactic and semantic structures are required for EAE models and should be explicitly combined to identify crucial context words for effective representations for EAE.

The question then arises: how can syntactic and semantic structures be combined to aid in learning effective representations for EAE? Pouran Ben Veyseh et al. (2020) propose using Graph Transformer Networks (GTN) (Yun et al., 2019) to facilitate syntax-semantic merging for EAE. GTNs allow for the combination of multiple input structures in two steps. The first step yields weighted sums of the input structures that serve as intermediate structures capable of capturing information from different input perspectives. In the second step, these intermediate structures are multiplied to produce the final structures. The aim here is to leverage multi-hop paths/connections between pairs of nodes/words to compute the importance score for the final structures.

Lastly, to further enhance the performance for EAE, a novel inductive bias is proposed for the GTN model. This is done to improve the model's

Passage: Saddam's family left that city three days ago.

Trigger identification
Q_1: Which word is the trigger word?
A_1: left

Trigger classification
Q_2: The trigger word is left $\langle pos \rangle$ 2 $\langle /pos \rangle$, movement: transport?
A_2: YES

Argument extraction
Q_3: left $\langle pos \rangle$ 2 $\langle /pos \rangle$. Movement:transport, time-within?
A_3: three days ago

Q_4: left $\langle pos \rangle$ 2 $\langle /pos \rangle$. Movement:transport, artifact?
A_4: Saddam's family

Q_5: left $\langle pos \rangle$ 2 $\langle /pos \rangle$. Movement:transport, destination?
A_5: NULL

...

Figure 8: An illustration and summary of the MQAEE framework (Li et al., 2020). In this context, the sentence under consideration serves as the passage. Each round consists of a question (represented as Qi) and a response (denoted as Ai). If a question doesn't have a corresponding answer, it's indicated by the term NULL.

generalization using the Information Bottleneck concept. Specifically, the combination of rich structures from syntax and semantics could enhance GTNs with a high capacity to encode detailed information in the input sentences. Given the generally small training datasets for EAE, GTN models might learn to retain all context information in the input sentences, including irrelevant information for EAE. This could lead to overfitting of GTNs on the training data. To tackle this, this study proposes treating the GTN model as an information bottleneck. In this way, the representations produced by GTNs are trained to not only achieve high performance for EAE prediction but also minimize the mutual information with the input sentences. An additional term in the overall loss function is introduced for this purpose: the mutual information between the generated representations of GTNs and the input sentences. This improves the generalization of GTNs for EAE. Comprehensive experiments on two benchmark datasets for EAE demonstrate that the proposed model can achieve leading performance for EAE.

3.4 Event Extraction as Multi-turn Question Answering

Li et al. (2020) introduce a novel method to solve event extraction via question answering. The system should be capable of identifying event triggers, their specific event types, and corresponding arguments with their roles. Consider an instance provided in Figure 8 where the Movement Transport event is activated by the trigger word "left", associated with three arguments: "Saddam's family" as Artifact, "that city" as Origin, and "three days ago" as Time-Within.

Typically, event extraction is bifurcated into two sections: trigger extraction and argument extraction, based on the standard Automatic Content Extraction (ACE) 2005 benchmark (Walker et al., 2006). Current methods to event extraction are majorly divided into two categories: (a) pipeline methods that carry out trigger extraction and argument extraction in separate stages (Liao and Grishman, 2010; Hong et al., 2011; Lu and Roth, 2012); (b) joint approaches that simultaneously perform all subtasks in a unified learning manner (Li et al., 2013; Nguyen et al., 2016; Liu et al., 2018).

These methods, regardless of being pipeline or joint, usually frame event extraction as classification tasks, by categorizing event triggers into predefined event types and subsequent event arguments into predetermined argument roles. However, this approach has two primary limitations. Firstly, the inability to explicitly model the semantics of these golden labels and to capture their rich interactions, which can be beneficial for event extraction. For instance, the event type Movement Transport can provide valuable supplements to the corresponding argument roles like Origin and Time-Within. Moreover, the identification of "Saddam's family" as an Artifact implies that "that city" might be an argument of the role Origin.

The second limitation is the lack of generalization ability. These classification-based methods cannot be extended to new event types or argument roles without additional annotations. A recent approach proposed by Huang et al. (2018) introduced a transfer learning architecture for zero-shot event extraction. Their methodology was to represent event mentions and event types (or arguments and argument roles) in a shared semantic space, and consider trigger (or argument) classification as a semantic matching problem. However, this approach struggles to capture full mention-type (or argument-role) interactions with only the final cosine similarity matching and heavily depends on structured features prone to error propagation.

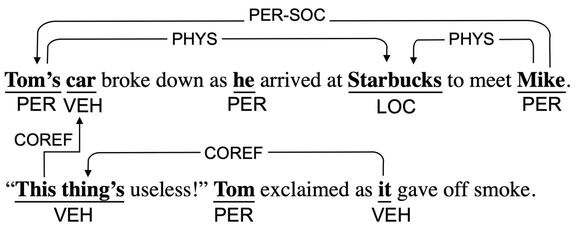

Figure 9: A textual excerpt showcasing the interplay among entities, relations, and coreference links. Certain relation and coreference connections are intentionally left out (Luan et al., 2019).

3.5 A General Framework for Information Extraction using Dynamic Span Graphs

A variety of Information Extraction (IE) tasks necessitate the identification and categorization of phrase spans, some of which may be nested. For instance, the process of entity recognition entails attributing an entity label to a phrase span. Meanwhile, Relation Extraction (RE) requires the assignment of a relation type between pairs of spans, and coreference resolution aims to consolidate spans that refer to the same entity into a single cluster. Consequently, one might anticipate that knowledge gained from one task could prove beneficial to another.

Historically, most IE research, such as that of Nadeau and Sekine (2007) and Chan and Roth (2011), utilized a pipeline approach, where entities were detected initially and then the recognized entity spans were employed for relation extraction and coreference resolution. To circumvent the cascading errors inherent to pipeline-style systems, recent studies have shifted their focus to the integration of different IE tasks. This includes the joint modeling of entities and relations (Miwa and Bansal, 2016), entities and coreferences (Hajishirzi et al., 2013), joint inference (Singh et al., 2013), or multi-task learning of entity/relation/coreference (Luan et al., 2018). Predominantly, these models depend on the first layer LSTM for sharing span representations between various tasks and are typically designed for specific domains.

Luan et al. (2019) present the Dynamic Graph IE (DYGIE) framework, a generalized structure for consolidating multiple information extraction tasks through shared span representations, which are honed by incorporating contextualized information from relations and coreferences. This framework has shown effectiveness across several domains, exhibiting the advantages of integrating a wider context learned from relation and coreference annotations.

An example illustrating the potential advantages of entity, relation, and coreference contexts is shown in Figure 1. Predicting the entity labels for "This thing" and "it" is impossible based solely on the within-sentence context. However, the antecedent "car" strongly indicates that these two entities fall under the VEH type. In a similar vein, the knowledge that Tom is located at Starbucks, and Mike is related to Tom, provides support for the notion that Mike is also at Starbucks.

DYGIE utilizes multi-task learning to recognize entities, relations, and coreferences by sharing span representations and by using dynamically constructed span graphs. The graph's nodes are dynamically chosen from a beam of high-confidence mentions, with edges weighted according to the confidence scores of relation types or coreferences. Unlike the multi-task methodology, which merely shares span representations from the local context (Luan et al., 2018), this framework leverages rich contextual span representations by propagating information through coreference and relation links. Distinct from previous BIO-based entity recognition systems (Lample et al., 2016) that assign a text span to a maximum of one entity, this framework enumerates and represents all possible spans to recognize entities that may overlap arbitrarily.

3.6 A Joint Neural Model for Information Extraction with Global Features

Traditional efforts in information extraction have typically adopted a pipeline approach (Liao and Grishman, 2010; Hong et al., 2011; Lu and Roth, 2012). Information Extraction (IE) is an intricate task, with the objective of pulling structured data from unstructured texts. This multifaceted process involves a variety of subtasks such as named, nominal, and pronominal mention extraction, entity linking, entity coreference resolution, relation extraction, event extraction, and event coreference resolution.

However, the typical pipeline method for IE often encounters the issue of error propagation and limits the interaction among the different components within the pipeline. This spurred researchers to look for solutions, and they proposed joint inference and joint modeling methods (Li et al., 2013; Nguyen et al., 2016; Liu et al., 2018), in an attempt to enhance local prediction. These methods offered

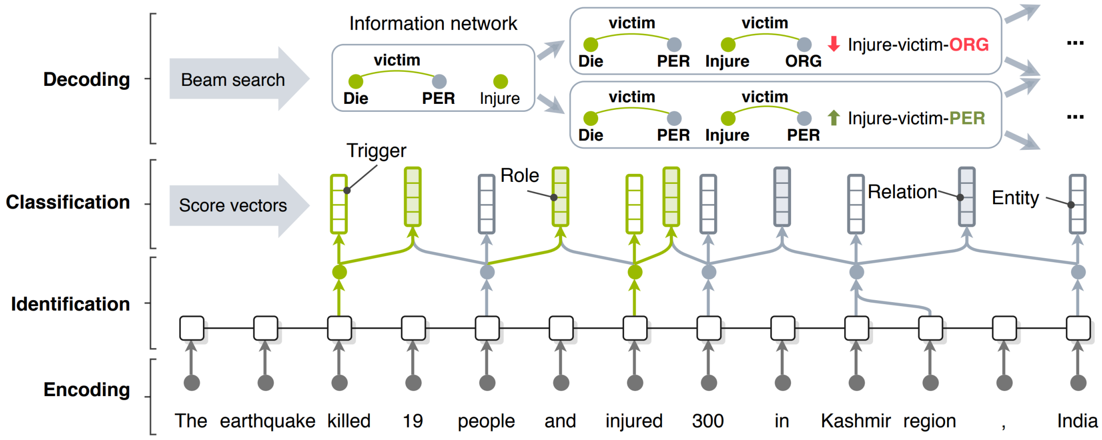

Figure 10: A depiction of the ONEIE framework (Lin et al., 2020), utilized for end-to-end joint information extraction during the testing phase. For simplicity's sake, not all pairwise connections are displayed.

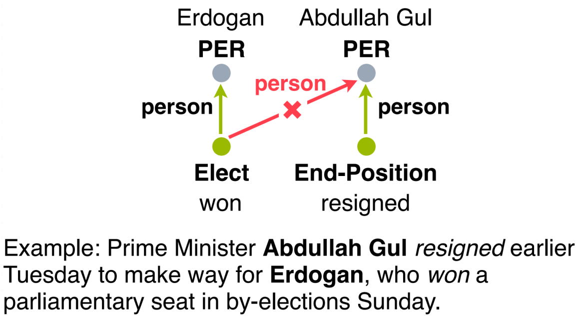

Figure 11: A common mistake made by local classifiers that lack global constraints. (Lin et al., 2020).

a better approach and saw significant development in their implementation and effectiveness, particularly with the rise of deep learning.

Due to the success of deep learning, neural models became a popular choice and were applied extensively across various IE subtasks. The increased application of neural models marked a turning point in the field of IE, changing how these tasks were approached and managed. More recently, efforts have been made to revisit global inference approaches, leading to the design of neural networks with embedding features. These new methods aim to jointly model multiple subtasks. However, these methods also have their shortcomings, as they tend to use separate local task-specific classifiers in their final layer. Moreover, they do not explicitly model the interdependencies among tasks and instances.

An example of this is seen in the case of a local argument role classifier predicting a redundant PERSON edge (see Figure 11. Ideally, models should avoid such mistakes, and they could do this if they had the ability to learn and leverage specific aspects of data, like the unusual occurrence of an ELECT event having two PERSON arguments.

To address this issue, a new joint neural framework, known as ONEIE (Lin et al., 2020), has been proposed to perform end-to-end IE with global constraints. Unlike traditional methods that use local classifiers to predict separate knowledge elements, ONEIE aims to extract a globally optimal information network for the input text. This process involves comparing candidate information networks during the decoding process, where not only are the individual label scores for each knowledge element considered, but also the cross-subtask and cross-instance interactions in the network.

The process of information extraction with ONEIE can be broken down into four primary steps: encoding, identification, classification, and decoding (see Figure 10). Initially, the input sentence is encoded using a pre-trained BERT encoder. Then, the entity mentions and event triggers within the sentence are identified. This is followed by the computation of the type label scores for all nodes and pairwise edges among them. Finally, during the decoding stage, possible information networks for the input sentence are explored using a beam search, and the information network with the highest global score is selected.

3.7 Cross-Task Instance Representation Interactions and Label Dependencies for Joint Information Extraction with Graph Convolutional Networks

Previous works such as DyGIE and OneIE introduce deep learning techniques to achieve state-of-the-art performance (Wadden et al., 2019; Lin et al.,

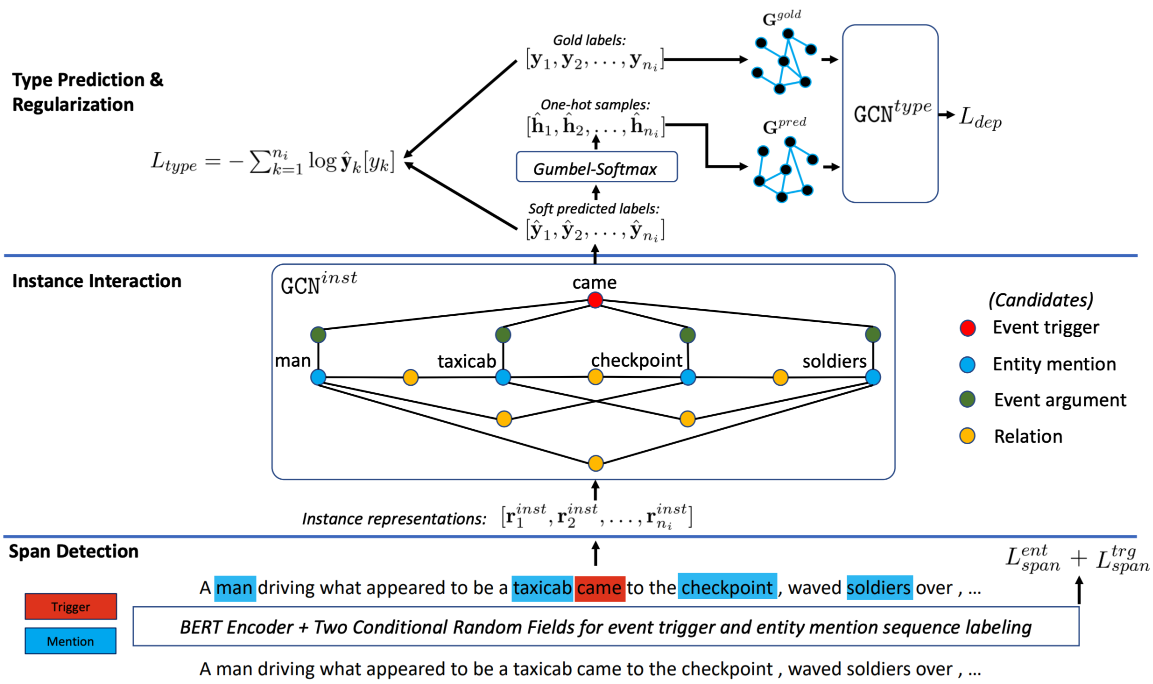

Figure 12: Overview of the FourIE framework (Nguyen et al., 2021a).

2020) for joint information extraction (Joint IE). However, two challenges remain, hindering further advancements in these models.

First, at the instance level, an essential aspect of deep learning models for joint IE involves the representation vectors of instances. These vectors are utilized to predict IE tasks within a given sentence, commonly known as predictive instance representations. In the case of joint four-task IE, we contend that there exist inter-dependencies between the predictive representation vectors of related instances from the four tasks. Incorporating these inter-dependencies into the model can enhance IE performance. For example, the entity type information encoded in the representation vector for an entity mention can influence the argument role captured by the representation vector for a related EAE instance (e.g., involving the same entity mention and an event trigger in the same sentence), and vice versa. However, previous studies on joint four-task IE have only computed predictive representation vectors for instances independently, using shared hidden vectors from a deep learning layer (Wadden et al., 2019; Lin et al., 2020). Although this shared mechanism partially captures the interaction between predictive representation vectors, it fails to explicitly model the connections between related instances from different tasks and integrate them into the representation learning process. To address this issue, (Nguyen et al., 2021a) propose a novel deep learning model called FourIE, which incorporates a graph structure to explicitly capture interactions between related instances from the four IE tasks within a sentence. Subsequently, a graph convolutional network (GCN) (Kipf and Welling, 2017; Nguyen and Grishman, 2018) is employed to enrich the representation vector of an instance by incorporating information from the related (neighboring) instances for IE.

Second, at the task level, existing joint four-task models for IE have only leveraged cross-task type dependencies during the decoding step to constrain predictions for a given sentence. This is achieved by manually converting the type dependency graphs of the sentence into global feature vectors for scoring predictions in the beam search-based decoding process (Lin et al., 2020). However, this knowledge derived from cross-task type dependencies does not contribute to the training process of the IE models. It is unfortunate since we anticipate that a deeper integration of this knowledge into the training process could offer valuable information to enhance representation learning for IE tasks. To tackle this, we propose utilizing the knowledge from cross-task type dependencies to provide an additional training signal for each sentence, directly supervising our joint four-task IE model. Specifically, our approach involves orga-

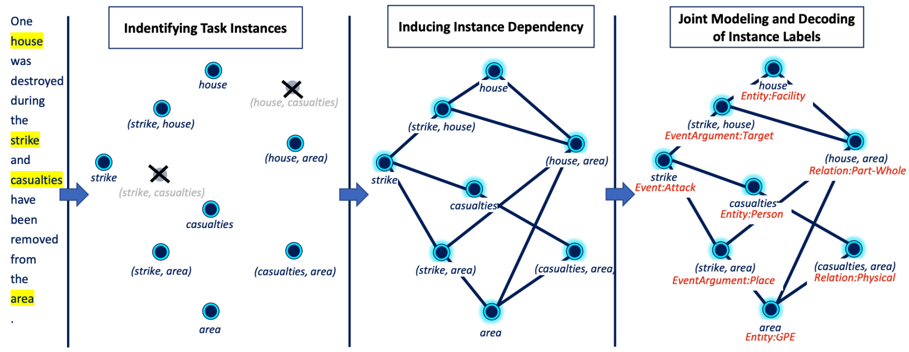

Figure 13: Overview of the three stages in GraphIE model (Nguyen et al., 2022a): i) identifying task instances, ii) inducing instance dependency, and iii) joint modeling and decoding of instance labels.

nizing the types expressed in a sentence for the four IE tasks into a dependency graph, which represents global type dependencies for the sentence. Therefore, for a joint model to perform well, the type dependency graph generated by its predictions should resemble the dependency graph obtained from the ideal types (i.e., a global type constraint on the predictions during the training step). To enforce this constraint, we introduce a novel regularization term into the training loss of our joint model. This term employs another GCN to learn representation vectors for both the predicted and ideal dependency graphs, promoting graph similarity. FourIE is the first work that incorporates global type dependencies as a regularization technique for joint IE models. An illustration of FourIE is provided in Figure 12

3.8 Joint Extraction of Entities, Relations, and Events via Modeling Inter-Instance and Inter-Label Dependencies

Joint Information Extraction - JointIE is a method that addresses the challenges of error propagation and dependency among prediction instances in ETD, EMR, EAE, and RE tasks (Wadden et al., 2019; Lin et al., 2020; Zhang and Ji, 2021; Nguyen et al., 2021a). To exploit instance dependency, previous studies like Wadden et al. (2019) and Lin et al. (2020) used a shared encoder to obtain representation vectors for different IE tasks. Later research attempted to capture dependency by connecting task instances with shared entity mentions or aligning instances with text spans on a semantic graph Zhang and Ji (2021); Nguyen et al. (2021a). However, these manual designs may not be optimal for representation learning in JointIE.

Apart from representation learning, prior work tends to factorize the joint distribution of labels in JointIE into individual distributions, limiting the utilization of label interactions across tasks (Lin et al., 2020; Zhang and Ji, 2021). Some approaches mitigate this by decoding labels using global features or encoding label interactions with consistency regularization over global dependency graphs (Nguyen et al., 2021a). However, these methods still assume factorization of the joint label distribution, failing to address the label dependency encoding issue. Recent attempts reformulated JointIE as a text generation problem, directly modeling the joint distribution of instance labels using pretrained seq2seq models like BART or T5 (Lewis et al., 2020; Raffel et al., 2020). Unfortunately, this approach relies on decoding task instances in a specific order, preventing later instances from correcting earlier predictions and leading to suboptimal performance.

In our approach (called GraphIE) (Nguyen et al., 2022a), we aim to overcome these challenges by inducing dependency between task instances in JointIE based on the data to enhance representation learning and model the joint distribution of labels (see Figure 13). We treat each task instance as a node in a fully connected dependency graph and learn the weights of the edges to capture the dependency level between corresponding instances. This approach differs from prior work that uses sparser dependency graphs with disconnected task instance pairs, thus limiting interaction exploration (Nguyen et al., 2021a; Zhang and Ji, 2021). We utilize Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) to enhance the representation of each instance node, incorporating information from all other nodes based on their dependency levels (Kipf and Welling, 2017; Nguyen and Gr-

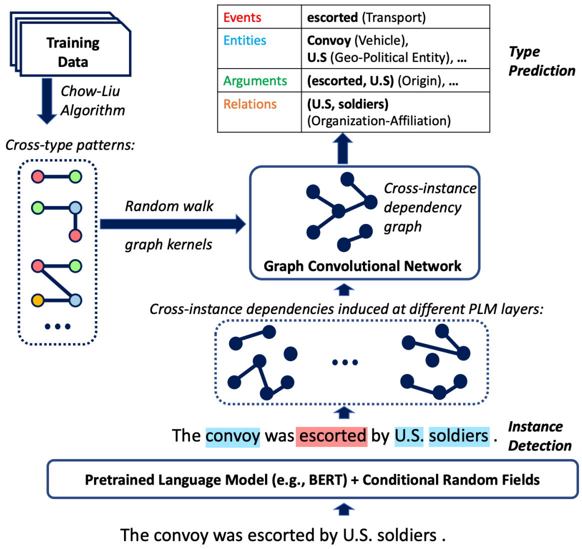

Figure 14: Overview of the DepIE framework for Joint IE (Nguyen et al., 2022b).

ishman, 2018). Subsequently, the improved instance representations and the induced dependency graph are used to estimate the joint distribution of instance labels using Conditional Random Fields (CRFs) (Lafferty et al., 2001). This formulation allows us to approximate the intractable joint likelihood of the ground-truth instance labels through Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2012), which transforms the maximization problem into nonlinear logistic regression to differentiate true labels from noise labels.

Finally, prior work in JointIE typically employed greedy or beam search for decoding instance labels, which is suboptimal due to its greedy nature. In our approach, we propose a novel decoding algorithm for JointIE using Simulated Annealing (SA) (Kirkpatrick et al., 1983), known for approximating global optima.

3.9 Learning Cross-Task Dependencies for Joint Extraction of Entities, Events, Event Arguments, and Relations

Capturing dependencies between the IE tasks is a crucial challenge for JointIE, including cross-instance and cross-type dependencies. Cross-instance dependencies involve instances referring to word spans for event triggers/entity mentions and their classification based on predefined information types. Previous JointIE models have enriched the representation of one instance by incorporating information from related instances in different IE tasks to aid type prediction (Lin et al., 2020; Nguyen et al., 2021a). Typically, creating dependency graphs between instances has been used to encode cross-instance dependencies and facilitate representation learning (Zhang and Ji, 2021; Nguyen et al., 2021a). However, manually designing these dependency graphs using heuristics, such as connecting instances that share an entity mention or event trigger, may not optimize performance for a specific dataset.

To address this limitation and enhance representation enrichment with information from related instances in JointIE, our approach (called DepIE) (Nguyen et al., 2022b) suggests automatically learning cross-instance dependency graphs from data (see Figure 14). We explore a fully connected graph connecting all task instances in a sentence, assigning a dependency weight to each edge to quantify their relatedness. Our method argues that dependency weights should be computed from multiple sources of information to generate optimal and comprehensive dependency graphs. Inspired by the encoding of linguistic structures in pre-trained language models (PLMs) like BERT (Devlin et al., 2019), we leverage instance representations from different layers of PLMs to calculate dependency weights. For each pair of instances in JointIE, their representation vectors at each layer of a PLM are utilized to produce layer-specific dependency weights. These weights are combined across layers to obtain an overall weight for the dependency graph. Enriched representations for the instances are then induced using Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) based on the computed cross-instance dependency graph.

In addition to cross-instance dependencies, cross-type dependencies highlight the co-occurrences or correlations between information types of different IE tasks within a sentence. For example, in the ACE 2005 dataset (Walker et al., 2006), a "Victim" argument for an "Attack" event is likely to be the same for a "Die" event in the same sentence. Previous JointIE models have incorporated cross-type dependencies either during decoding, by forming global type patterns/graphs to constrain type predictions, or during training, by forming type dependency graphs to aid consistency regularization of golden and predicted types (Lin et al., 2020; Nguyen et al., 2021a). However, similar to cross-instance dependencies, previous work manually designed dependency graphs

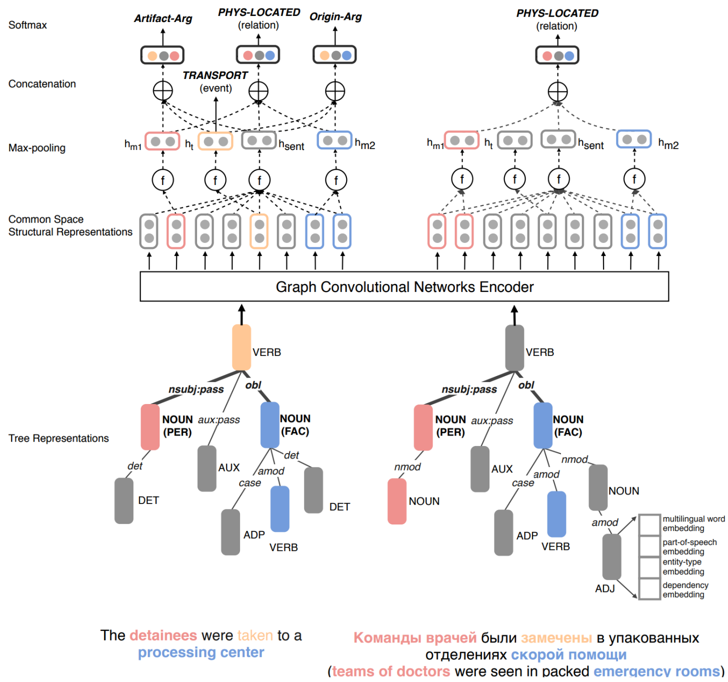

Figure 15: Multilingual common semantic space and cross-lingual structure transfer (Subburathinam et al., 2019).

between information types, often linking types involved in the same instance for a specific IE task (Nguyen et al., 2021a). This manual design approach may overlook crucial cross-type patterns, leading to suboptimal performance in JointIE.

To address this limitation and better support type predictions in JointIE, our proposed method further learns cross-type dependencies/patterns from data. Each information type in the IE tasks is viewed as a binary random variable that signifies its presence or absence in a sentence. This formulation enables us to employ Bayesian structure learning algorithms, such as the Chow-Liu algorithm (Chow and Liu, 1968), which measures mutual information between types in the training data to learn a first-order dependency tree approximating the underlying joint distribution of the information types for JointIE. The resulting Chow-Liu tree captures induced dependencies between information types and is used to generate global cross-type patterns.

To incorporate the learned cross-type dependencies into the JointIE model, our objective is to leverage these patterns as additional features to further enhance the GCN-induced representations for type prediction. We treat the induced cross-type patterns as anchor knowledge that governs the information types, representations, and dependencies of IE instances in a sentence, ensuring consistency and improving predictions for JointIE. For each learned cross-type pattern, we compute a similarity score between the computed cross-instance dependency graph for an input sentence and the cross-type pattern. This similarity score is then included in the representations of the instances to predict types. To achieve this, we propose leveraging random walk graph kernels, which count common random walks on the graphs, to facilitate similarity computation between the cross-instance dependency graph and the cross-type pattern. This enriches the representations for JointIE.

4 Cross-lingual Transfer Learning for IE

4.1 Cross-lingual Structure Transfer for Relation and Event Extraction

Gold-standard annotations for relation and event extraction are available for only a limited number of languages (Walker et al., 2006; Getman et al., 2018). Obtaining these annotations is more expensive compared to other information extraction tasks like name tagging because they involve structured data and require a wide range of labels.

Recent research (Lin et al., 2017) has discovered that relational facts in languages often follow identifiable patterns. These patterns can be leveraged to enhance relation extraction by considering language-universal features related to the identification and classification of relation and event arguments. Examples of such features include language-universal POS tagging, universal dependency parsing, entity extraction, and multilingual word embeddings.

By utilizing these language-universal representations, we can project entity mentions, event triggers, and their contexts into a unified multilingual space. Unlike previous methods that rely on linear mappings or canonical correlation analysis, Subburathinam et al. (2019) introduce a novel approach that converts text data into structured representations derived from universal dependency parses and enriched with distributional information (see Figure 15). This allows us to capture individual entities, relations, and events across multiple languages and share structural representations.

Next, Subburathinam et al. (2019) develop a novel framework for cross-lingual structure transfer learning. This framework enables us to project training data from a source language and test data from a target language into a common semantic space. By training a relation or event extractor using the source language annotations, we can apply the resulting extractor to texts in the target language. The authors employ graph convolutional networks (GCN) (Kipf and Welling, 2017) to encode graph structures in the input data, generating latent representations for entities and words. Unlike other encoders, GCN captures more comprehensive contextual information from dependency parses by considering all parse tree neighbors of each word, rather than just the child nodes. Using this shared encoder, we consider relation extraction and event argument role labeling as mappings from the latent space to relation type and event type with their respective argument roles.

4.2 Crosslingual Transfer Learning for Relation and Event Extraction via Word Category and Class Alignments

Previous works on crosslingual Relation and Event Extraction (crosslingual REE) predominantly use multilingual word embeddings such as MUSE (Joulin et al., 2018; Subburathinam et al., 2019) or multilingual pre-trained language models like multilingual BERT (Devlin et al., 2019; M'hamdi et al., 2019) to form crosslingual representation vectors for REE.

Still, past efforts on crosslingual REE encounter the issue of monolingual bias due to training models solely on source language data, resulting in suboptimal crosslingual performance. One potential remedy for this issue might be language adversarial training (Chen et al., 2019; Huang et al., 2019; Lange et al., 2020) where unlabeled target language data is utilized to assist crosslingual representations through tricking a language discriminator. The fundamental idea behind this method is to bring closer the representation vectors for sentences in the source and target languages. However, an inherent flaw of language adversarial training is its inability to condition on classes/types of examples during alignment, leading to potential mismatches in class alignment and decreased model performance.

To address this, Nguyen et al. (2021c) suggest to employ crosslingual alignment techniques that explicitly take into account class information of REE tasks to improve representation alignment and learning (see Figure 16). The principal intuition is that the semantics of classes in REE tasks (such as the event type of Attack in event extraction) are generally consistent across languages and can be used as anchors to connect representation vectors for examples in various languages. In this approach, two semantic representation vectors for each class in an REE task are generated based on representation vectors of examples in either source or target language. Subsequently, the representation vectors of the same class are adjusted to match each other, acting as a class-aware crosslingual alignment mechanism for source and target examples. Multilingual BERT (mBERT) is used to attain same-space representations for examples in both source and target languages to facilitate this alignment process.

Along with class semantics, a proposal is also made to leverage universal parts of speech and dependency relations in parsing trees to improve the cross-lingual alignment for representation vectors in REE. Given that these universal word categories have been consistently annotated for over 100 languages (Zeman et al., 2019) and can be produced with high accuracy using existing toolkits, they are expected to provide valuable anchor knowledge for cross-lingual representation learning.

However, a potential issue arises when calculating word category representations via contextualized representations of examples due to the preservation of context word information in representations for word categories that could introduce noise and obstruct the representation alignment. To mitigate this, an adversarial training model is proposed that aims to filter out context informa-

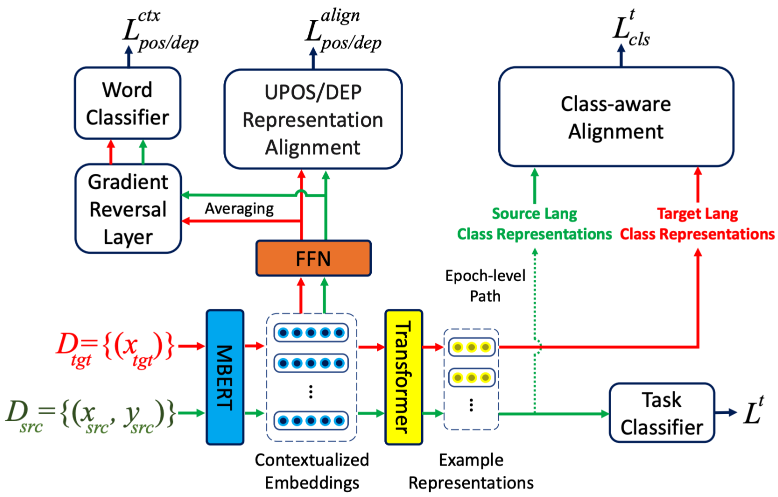

Figure 16: Overall architecture of the proposed models for RE, EAE. For ED, example representations are the contextualized embeddings (Nguyen et al., 2021c).

tion from word category representations. This is accomplished by employing the Gradient Reversal Layer (Ganin and Lempitsky, 2015) to prevent word category representations from being able to identify the context words in the original examples. It is anticipated that this filtering mechanism can enhance the purity of the word category representations, thereby providing suitable inputs for the alignment process for enhanced representation learning.

4.3 Cross-Lingual Event Detection via Optimized Adversarial Training

Cross-Lingual Event Detection (CLED) suggests the process of generating models that perform Event Detection (ED) effectively across multilingual data, introducing unique difficulties. One such challenge is the variations in trigger words between languages, for example, the inconsistencies in verb tenses. Proper verb management is crucial in ED as event triggers are often linked to sentence verbs. A study by Majewska et al. (2021) attempted to resolve this issue by infusing external verb knowledge during model training. Another issue specific to CLED is the presence of triggers that have diverging meanings in different languages. For instance, in Spanish, the term “juicio” can mean “judgement” or “trial” in English, depending on the context.

A promising strategy for developing a crosslingual model involves employing transfer learning, which applies the performance of a model trained in one language to a second, target language. The overarching concept is to use high-quality annotated data from a resource-rich language to train a model, allowing it to grasp language-agnostic aspects of the task - in this case, ED - and therefore perform effectively on a second language’s text. Previous works using transfer learning for CLED leveraged pre-trained Multilingual Language Models (MLMs), like multilingual BERT (mBERT) (Devlin et al., 2019), benefiting from their inherent language-agnostic attributes. However, these models could be improved further as they sometimes have difficulty handling complex cases unique to cross-lingual contexts. It is noted that previous CLED attempts do not make use of the copious amounts of unlabeled data available, even though MLMs are trained using vast quantities of it. The belief is that incorporating unlabeled target-language data into the training process should enhance language context understanding, helping to address issues such as verb variations and multiple meanings.

Consequently, Guzman-Nateras et al. (2022) propose to use Adversarial Language Adaptation (ALA) (Joty et al., 2017; Chen et al., 2018) to train a CLED model. The core concept involves generating language-agnostic representations that do not reflect the language but remain useful for the ED task. Unlabeled data from both source and target languages are used to train a Language Discriminator (LD) network that discerns between the two languages. The adversarial aspect arises from the encoder and discriminator being trained with conflicting goals. As the LD improves in distinguishing languages, the encoder strives to produce more language-agnostic representations in an effort to deceive the LD. To the best of anyone's knowledge, this is the first time ALA has been proposed for the CLED task.

However, unlike previous ALA applications where all unlabeled samples are treated equally, it is acknowledged that this approach is not ideal. Some samples are more informative for the discriminator than others. Ideally, the LD should be exposed to samples that allow it to understand the nuanced differences between the source and target languages, rather than relying solely on syntactic differences. Furthermore, within the context of ED, it would be beneficial for the LD to train with event-containing examples rather than non-event samples, allowing the presence of an event to be integrated into the generated representations.

Therefore, it is suggested to refine the adversarial training process by focusing on the most informative examples and disregarding the less helpful ones. The rationale behind identifying samples as more informative for CLED is twofold: Firstly, if the LD is exposed to examples that are too different, the discrimination task becomes too straightforward. The aim is for the LD to understand the nuanced distinction between source and target languages, which, in turn, enhances the language-invariance of the encoder's representations. So, presenting the LD with examples possessing similar contextual semantics, i.e., similar contextualized representations, is suggested. Secondly, sentences that contain events should give an ED system more task-related information compared to non-event samples. It is therefore argued that sentences containing events should have a higher probability of being chosen for ALA training.

Keeping these insights in mind, Optimal Transport (OT) (Villani et al., 2009) is proposed as a practical solution to combine both the similarity between sample representations and the likelihood of the samples containing an event within one framework. Thus, the process of sample selection is seen as an OT problem in which the best alignment between source and target language samples is sought.

5 Conclusions and Future Work

This work provides a comprehensive examination of the key research directions in Multilingual Information Extraction (Multilingual IE), an essential area of study within the broader landscape of Natural Language Processing. Our investigation demonstrates the pivotal role of Multilingual IE in meeting the escalating demand for efficient communication and understanding across thousands of languages worldwide. The study's focus encompasses three major research trajectories: enhancing Multilingual IE upstream models, developing language-agnostic downstream models, and advancing cross-lingual transfer learning methods for situations with scarce training data. Despite the remarkable advancements made in these areas, the paper highlights enduring challenges, such as the model size and memory use, error propagation in pipeline-based models, and monolingual bias in crosslingual transfer learning. Addressing these challenges presents an exciting frontier in the pursuit of democratizing communication and information access in our linguistically diverse world.

The future of Multilingual IE research is brimming with opportunities. One of the critical areas requiring further attention is the development of efficient model compression techniques for upstream models. Reducing model size without compromising performance will make these models more accessible and deployable across diverse computational environments. Secondly, further exploration into JointIE models is necessary to fully harness the language differences and similarities for improved multilingual training and model generalization. The mitigation of error propagation in these models will significantly enhance the quality of information extracted. Finally, the improvement of cross-lingual transfer learning approaches warrants more in-depth research, especially in scenarios with limited training data. More robust techniques need to be developed to combat the monolingual bias and enable seamless portability of models across languages. The successful addressal of these research directions will contribute significantly to the ongoing efforts of democratizing access to information and facilitating global communication, irrespective of the language barriers. The essence of future work in Multilingual IE is to continue the pursuit of making the rich linguistic diversity of our world a connecting bridge rather than a dividing barrier.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. NoSta-D named entity annotation for German: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland. European Language Resources Association (ELRA).

Steven Bethard, Philip Ogren, and Lee Becker. 2014. ClearTK 2.0: Design patterns for machine learning in UIMA. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3289–3293, Reykjavik, Iceland. European Language Resources Association (ELRA).

J Bloch. 2008. Effective java. addison-wesley professional.

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.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy. Association for Computational Linguistics.

Xilun Chen and Claire Cardie. 2018. Unsupervised multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270, Brussels, Belgium. Association for Computational Linguistics.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

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.

CKCN Chow and Cong Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467.

James Clarke, Vivek Srikumar, Mark Sammons, and Dan Roth. 2012. An NLP curator (or: How I learned to stop worrying and love NLP pipelines). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3276–3283, Istanbul, Turkey. European Language Resources Association (ELRA).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 168–175.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland. European Language Resources Association (ELRA).

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.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

David Ferrucci and Adam Lally. 2004. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Jeremy Getman, Joe Ellis, Stephanie Strassel, Zhiyi Song, and Jennifer Tracey. 2018. Laying the groundwork for knowledge base population: Nine years of linguistic resources for TAC KBP. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Iryna Gurevych, Max Mühlhäuser, Christof Müller, Jürgen Steimle, Markus Weimer, and Torsten Zesch. 2007. Darmstadt knowledge processing repository based on uima. In *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology, Tübingen, Germany*, page 89.

Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2).

Luis Guzman-Nateras, Minh Van Nguyen, and Thien Nguyen. 2022. Cross-lingual event detection via optimized adversarial training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5588–5599, Seattle, United States. Association for Computational Linguistics.

Udo Hahn, Franz Matthies, Erik Faessler, and Johannes Hellrich. 2016. UIMA-based JCoRe 2.0 goes GitHub and maven central — state-of-the-art software resource engineering and distribution of NLP pipelines. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2502–2509, Portorož, Slovenia. European Language Resources Association (ELRA).

Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 289–299, Seattle, Washington, USA. Association for Computational Linguistics.

Geert Heyman, Bregt Verreet, Ivan Vulić, and Marie-Francine Moens. 2019. Learning unsupervised multilingual word embeddings with incremental multilingual hubs. 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 1890–1902, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Portland, Oregon, USA. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Lifu Huang, Heng Ji, and Jonathan May. 2019. Cross-lingual multi-level adversarial transfer to enhance low-resource name tagging. 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 3823–3833, Minneapolis, Minnesota. Association for Computational Linguistics.

Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. 2017. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 226–237, Vancouver, Canada. Association for Computational Linguistics.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science*, 220(4598):671–680.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Viet Lai, Hieu Man, Linh Ngo, Franck Dernoncourt, and Thien Nguyen. 2022a. Multilingual SubEvent relation extraction: A novel dataset and structure induction method. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5559–5570, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5405–5411, Online. Association for Computational Linguistics.

Viet Dac Lai, Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2022b. MECI: A multilingual dataset for event causality identification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2346–2356, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

Lukas Lange, Anastasiia Iurshina, Heike Adel, and Jannik Strötgen. 2020. Adversarial alignment of multilingual models for extracting temporal expressions from text. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 103–109, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Neural relation extraction with multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34–43, Vancouver, Canada. Association for Computational Linguistics.

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*, pages 7999–8009, Online. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 835–844, Jeju Island, Korea. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. 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.

Olga Majewska, Ivan Vulić, Goran Glavaš, Edoardo Maria Ponti, and Anna Korhonen. 2021. Verb knowledge injection for multilingual event processing. 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 6952–6969, Online. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Meryem M'hamdi, Marjorie Freedman, and Jonathan May. 2019. Contextualized cross-lingual event trigger extraction with minimal resources. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 656–665, Hong Kong, China. Association for Computational Linguistics.

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.

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.

Kanthashree Mysore Sathyendra, Samridhi Choudhary, and Leah Nicolich-Henkin. 2020. Extreme model compression for on-device natural language understanding. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 160–171, Online. International Committee on Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

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, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021b. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, 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. 2021c. 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 Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

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. Event detection and domain adaptation with 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 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 886–891, Austin, Texas. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

Geondo Park, Gyeongman Kim, and Eunho Yang. 2021. Distilling linguistic context for language model compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 364–378, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anthony Patricio. 2009. Why this project is successful?

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Javid Ebrahimi, Franck Dernoncourt, and Thien Nguyen. 2022a. MEE: A novel multilingual event extraction dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9603–9613, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Nguyen. 2022b. MINION: a large-scale and diverse dataset for multilingual event detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2286–2299, Seattle, United States. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4393–4399, Florence, Italy. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 1–6.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland. 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.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent 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.

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.

Amir Veyseh, Franck Dernoncourt, My Thai, Dejing Dou, and Thien Nguyen. 2020. Multi-view consistency for relation extraction via mutual information and structure prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9106–9113.

Cédric Villani et al. 2009. *Optimal transport: old and new*, volume 338. Springer.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. In *Technical report, Linguistic Data Consortium*.

Hanna M Wallach. 2004. Conditional random fields: An introduction. *Technical Reports (CIS)*, page 22.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. HMEAE: Hierarchical modular event argument 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 5777–5783, Hong Kong, China. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0. *Linguistic Data Consortium*.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems*, 32.

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.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori

Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Manying Zhang, and Hanzhi Zhu. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

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.

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.