TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science

Institute of Artificial Intelligence
Chair of Machine Learning for Computer Vision

Master Thesis

Formalization of natural language processing tasks as optimization problems

Oleh Shkalikov

Born on: 28th February 2001 in Mariupol, Ukraine
Matriculation number: 5102818

5th December 2024

First referee
Prof. Dr. Bjoern Andres

Second referee
Prof. Dr. Simon Razniewski

Supervisor
Jannik Irmai

Statement of authorship

I hereby certify that I have authored this document entitled *Formalization of natural language processing tasks as optimization problems* independently and without undue assistance from third parties. No other than the resources and references indicated in this document have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the intellectual preparation of the present document. I am aware that violations of this declaration may lead to subsequent withdrawal of the academic degree.

Dresden, 5th December 2024

Oleh Shkalikov

TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Artificial Intelligence

Chair of Machine Learning for Computer Vision

Abstract

High-resource languages benefit from abundant data, allowing named entity recognition tasks to achieve high accuracy. However, low-resource languages and specialized domains face challenges due to limited training data. Current multilingual capabilities of models are insufficient for low-resource languages, prompting ongoing research. Approaches to address NER in these contexts are split into model-based and data-based, with projection-based methods being a significant subset of the latter. These methods involve translating sentences into high-resource languages, labeling them, and projecting labels back to the original. Despite many proposals for projection, this step hasn't been explicitly formalized as an optimization problem, often relying on heuristics. This study addresses this gap by formulating the projection step as an integer linear programming problem within the cross-lingual NER pipeline. This ILP approach optimizes matching of source to target entities using varied cost strategies: alignment-based, NER-model-based, and translation-based scores,—integrated to mitigate individual limitations. While the ILP problem's complexity remains an open question, a proposed greedy algorithm performs effectively in practice. Evaluations using Europarl-NER and MasakhaNER2 datasets show our ILP method excels over heuristic and model transfer techniques, especially with fused scores.

Contents

Symbols and Acronyms

ILP	Integer linear programming
NER	Named entity recognition
XLNER	Cross-lingual named entity recognition
MaxIS	Maximum independent set
MaxWIS	Maximum weight independent set

1 Introduction

Named entity recognition (NER) is one of the fundamental problems in natural language processing. It is important in itself, but it also plays a crucial role in information extraction pipelines, for example for knowledge graph construction [Wei+21; ZGN22]. Given a sentence and a set of classes, the goal is to predict a class for every word and then aggregate entities, which are defined as continuous sequences of words that belong to a single object. Typically, this aggregation is accomplished using a specialized labeling format called IOB2, where the label *O* is assigned to all words that do not belong to any class. For the first word of an entity with class C, the label *B-C* is assigned, and for subsequent words within that entity, the label *I-C* is assigned. An example of such labeling is provided in Figure 1.1.

Figure 1.1: Example of NER labeling in IOB2 format

To address this problem, there are currently two main classes of models: pretrained encoder-only Transformers [Vas17], such as BERT [Dev+19], and autoregressive large language models (LLMs) [Bro+20; Zho+23]. While the latter formulates NER as a generation problem within the context of general question answering, the former requires specially labeled datasets for training.

1.1 Motivation

For high-resource languages for which a lot of data is available, the NER problem can be addressed with a high degree of quality. However, challenges arise when dealing with low-resource languages or specialized domains where training data is limited. The multilingual capabilities of large language models, particularly their performance on low-resource languages, are limited and currently an active area of research [LMF24]. A multitude of approaches have been proposed to tackle the NER problem for low-resource languages, which can be categorized into two groups: model-based and data-based methods.

An important subset of the latter category is known as projection-based methods, which decompose the NER labeling problem into three steps: translating to high-resource languages, labeling the translated sentence, and projecting the labels back onto the original sentence. While many methods for the projection step have been proposed, the projection step itself has not been explicitly formalized as an optimization problem and, as a result, has not been analyzed. Consequently, some of these methods rely solely on heuristics. This highlights the necessity of formulating the projection step as an integer linear optimization problem and evaluating whether such a formulation is sensible from an application standpoint and whether it can help enhance the performance of the projection step.

1.2 Goal

The primary goal of this thesis is to formulate the projection step of the Cross-lingual named entity recognition (XLNER) pipeline as an Integer linear programming (ILP). This entails a comprehensive analysis of the problem, the establishment of relationships and connections between existing methods and the proposed formulation, and an evaluation of the proposed formalization in comparison to existing methods that have demonstrated consistent results.

1.3 Structure of the Work

This work consists of five chapters and two appendices. Chapter 2 provides an overview of existing XLNER pipelines, with a particular focus on projection-based methods. In Chapter 3, the projection step is formulated as an ILP problem, various matching costs and target candidate extraction methods are proposed, and insights regarding the complexity of the problem are discussed, alongside with an approximate greedy algorithm aimed to efficiently solve this problem. Chapter 4 evaluates the proposed formulation in isolation, focusing solely on the projection step, as well as within the full XLNER pipeline in intrinsic settings using the MasakhaNER2 dataset. Chapter 5 presents the conclusions derived from this work and suggests directions for further research.

Appendix A includes the runtimes for certain experiments, which may be useful for comparing different methods, while Appendix B demonstrates that the generalized version of the proposed ILP problem is NP-hard.

2 Background

Before proceeding to the description of our method, it is essential to understand methods for cross-lingual named entity recognition that have been already proposed. This chapter will provide an overview of several significant results and approaches in this field. However, this list is not exhaustive, as XLNER remains an active area of research.

Overall, all approaches can be categorized into two major groups: model-based and data-based methodologies.

2.1 Model transfer

Currently, there are two main types of models used for the NER problem. Both types are based on the Transformer [Vas17] architecture but leverage different components of it.

Encoder-only models, such as BERT [Dev+19], are pretrained in unsupervised settings using large volumes of text, which enhances their understanding of language. Following this pretraining, these models are fine-tuned for the NER task, requiring a labeled dataset for this process.

Conversely, decoder-only models, such as LLMs like GPT [Bro+20], are pretrained in unsupervised settings on even larger datasets. NER tasks can be performed by these models in a generative context, utilizing both simple prompting as well advanced formulation, e. g. as a task of code generation, as in GoLLIE [Sai+24], thereby eliminating the necessity for a labeled NER dataset. Additionally, a notable advantage of decoder-only models is their flexibility, i.e. they are not restricted to a predefined set of classes, unlike encoder-only models, which can only predict classes that were included in the training dataset.

However, due to their autoregressive nature, large language models require significant computational resources to perform the NER task, as they necessitate several forward passes of the model. In contrast, encoder-only models accomplish this in a single forward pass. To address the issue of resource requirements, some researchers have proposed distilling a large LLM into smaller LLMs, e. g. UniversalNER [Zho+23], or even into encoder-only models [HTC24].

In the context of cross-lingual NER, both approaches leverage the model's ability to transfer knowledge across languages. The general concept is as follows: models are pretrained on datasets containing multiple languages, thereby developing an understanding of each language. Subsequently, if a model is trained as in the case of BERT-like architectures, or is capable, as is the case of large language models, of addressing the NER problem in one language it can subsequently transfer this knowledge to tackle NER in other languages. Notable examples of such encoder-only models include mBERT [Dev+19], XLM-RoBERTa [Con+20], and MDeBERTa [He+20; HGC21]. Additionally, the pretraining datasets of all LLMs typically [Tou+23] contain data in multiple languages.

It has been demonstrated by [Tor+23] that when the language of the dataset used for the fine-tuning of an encoder-only model originates from the same language family as the target language, the performance of NER in the target language is likely to be higher compared to situations where the dataset is sourced from a different language family. Unfortunately, for low-resource languages such as Upper Sorbian, the languages within the same family also lack sufficient labeled data. A similar situation arises in specific domains, where labeled data is predominantly available only in English. The multilingual capability of LLMs for low-resource languages is also limited, and this topic constitutes an active area of research [LMF24].

Thus, despite the fact that model transfer demonstrates favorable results in some scenarios [GAR22], the issues associated with these methods underscore the necessity for alternative approaches to XLNER.

2.2 Data-based methods

Another significant group of approaches to cross-lingual named entity recognition consists of methods aimed at generating labeled datasets in the target language, referred to as data-based methods. While some approaches generate entirely artificial data, such as MulDA [Liu+21], in which the authors propose using a language model (LSTM [HS97] or mBART [Liu+20]) to generate labelled text in the target language by inserting labels in IOB2 format before the words with corresponding labels, the most compelling approaches not only generate text but also incorporate the labeling of the desired text in the target language.

The overall pipeline of such XLNER approaches is divided into three steps: the translation of the input sentence, referred to as the target sentence, into a high-resource language; the application of a NER model to the translated sentence, known as the source sentence; and the projection of entities from the source sentence back onto the sentence in the target language.

The NER entities identified by a NER model in the source sentence are referred to as source entities.

Definition 2.1 (Source entity)**.** Let $s^{src} = \left(s_1^{src}, \ldots, s_m^{tgt}\right)$ be a source sentence consisting of n words and L – a fixed set of classes. The source entity $p^{src} = (i_{p^{src}}, j_{p^{src}}, l_{p^{src}})$ is defined as a continuous subrange of words within the sentence, that corresponds to entity predicted by a NER model. It is characterized by the index $i_{p^{src}} \in \{1, \ldots, n\}$ of the first word, the index $j_{p^{src}} \in \{1, \ldots, n\}$ of the last word and the class $l_{p^{src}} \in L$.

This type of XLNER pipeline is called a projection-based pipeline. The main idea is that after translation, a sentence is obtained in a high-resource language for which a labeled dataset is likely available, therefore it is possible to train a model to perform the desired NER task.

While state-of-the-art transformer-based models, such as NLLB-200 [Tea+22] and DeBERTa [He+20; HGC21], are utilized for translation and source NER labelling, there are numerous methods to perform the final projection step. This work will propose a formulation of the projection step as an integer linear optimization problem, thereby necessitating a description of the existing projection methods. The first major group of projection methods consists of those based on back-translation, which involves translating labeled source sentences or their substrings back to the target language while preserving the labels.

CROP [Yan+22] surrounds each source entity with a special symbol `__SL0Tn__`, where n represents an index corresponding to the entity class. Subsequently, this slotted sentence is passed to a fine-tuned translation model designed to translate such sentences, resulting in a sentence in the target language. Projection occurs in the following manner: for each substring surrounded by slots with the same index, CROP searches for it in the original target sentence. If it is found, the class corresponding to the index of the slot is assigned to the identified substring; otherwise, no projection occurs. A significant weakness of CROP is that the back-translated sentence often differs from the original sentence, leading to low recall and the omission of many entities.

EasyProject [Che+23] also employs the insertion of special markers, specifically square brackets, which here remain consistent across different entity types, surrounding all source entities before passing the marked sentence into the translation model. The model then translates the entire sentence with the markers and also translates each source entity independently. Subsequently, fuzzy string matching is utilized to project labels: for each substring of the back-translated sentence that is surrounded by markers, the method identifies the back-translated source entity with the highest fuzzy matching string score and assigns the corresponding label to the substring. Unfortunately, this method also suffers from translation errors induced by the insertion of special symbols, as well as the fact that translating source entities independently, without context, often results in lower-quality translations. By its design, this method is incapable of projecting labels onto the original sentence and can therefore only be employed to generate a labeled dataset in the target language based on the labeled dataset in the source language.

CODEC [Le+24] aims to address the issue that the back-translated and the original target sentences differ by employing constrained decoding. This method similarly encloses every entity with special markers that corresponds to the class; however, it utilizes guided decoding (for which the authors propose an optimized version specifically for this task) to ensure that the back-translated sentence completely matches the original target sentence, except for the markers added around each entity. Nevertheless, since translation models remain imperfect and the insertion of markers impacts their performance, this method also exhibits limitations.

CLaP [Par+24] seeks to address the issue of independent source entity translation encountered in EasyProject by proposing the use of a LLM as a contextualized translator instead of employing a conventional translation model like NLLB-200. In this approach, the LLM is tasked with translating each source entity into the target language while being

provided with the entire source sentence as context. Initially, this method was proposed to generate a labeled dataset in the target language based on a labeled dataset in the source language. However, with the application of guided decoding that constrains the translation of source entities to be substrings of the original sentence, it can also be utilized for projection onto the original target sentence. The primary limitation of this approach lies in the language coverage of openly available LLMs, which typically [Tou+23] encompasses fewer than 200 languages, compared to NLLB-200, along with the fact that translation quality is still not perfect.

T-Projection [GAR23] employs a fine-tuned mT5 [Xue+21] model with beam search to generate substrings of the target sentence that serve as potential candidates for matching with source entities. It then computes the NMTScore [VS22], which represents the likelihood that the given source entity translates to the selected candidate, ultimately choosing the candidate with the highest NMTScore to project the source entity onto it. The primary drawbacks of this approach include the limitation of the mT5 model in terms of the number of languages it supports, the necessity for the model to be trained, and the significant computational resources required to generate just potential projection candidates. Additionally, the NMTScore is not without its flaws, as it is dependent on the quality of the underlying translation model.

As an alternative to translation-based projection, TransFusion [CSR23] proposes the use of specifically trained models to perform projection. The authors propose two types of fusion models capable of projecting entities: decoder-only large language models and encoder-only models. In the first case, the translated sentence in the source language and the original target sentence are passed to the LLM, which is then asked to perform source NER labeling and projection simultaneously by outputting labelling for both sentences. For the encoder-only model, every source entity is enclosed with XML tags corresponding to classes, and this is concatenated with the original target sentence before being passed to the encoder-only model. And then only the output that corresponds to the target sentence is considered. The main limitation of this approach is that it requires a parallel labeled dataset for training such models, which does not exist, which motivates the problem of XLNER. To address this fundamental issue, the authors employ EasyProject to generate such a dataset based on a labeled dataset in the source language. However, as EasyProject introduces errors, models trained on this generated data are also affected by these inaccuracies.

The final type of projection approaches is based on word-to-word alignments, which indicate whether a word in the source sentence corresponds to a word in the target sentence. The most effective methods for computing such alignments today involve neural network-based aligners, such as AWESoME [DN21] and SimAlign [Jal+20]. Here and in the following sections, alignments will be represented by an alignment matrix a_{kl}, where each value is binary, indicating whether a word from the source sentence with index k is aligned to a word in the target sentence with index l.

The core idea of this projection framework is rooted in the property that if words are aligned with one another, they should share the same label. However, since word-to-word alignments generated by models are not yet ideal, this approach encounters three primary challenges: split annotations, annotation collisions, and incorrect alignments. Split annotations refer to a situation in which a single source entity corresponds to multiple words from different parts of a target sentence, contrary to the expectation of a single continuous range of words.

Algorithm 1 : Algorithm that merges ranges that are separated by d non-aligned to the source entity words

Data : C – set of ranges of target words, a_{kl} – word-to-word alignments, e – index of the first word of the source entity, parameters $d \in \mathbb{N}$, only_i $\in \{True, False\}$
Result : $\hat{C}$ – set of target word ranges where all ranges with at most d non-aligned to the source words have been merged

$\mathcal{C} \leftarrow$ sort ranges C by the index of the first word ;
$\hat{C} \leftarrow \{\mathcal{C}[0]\}$;
$\hat{o} \leftarrow o_{\mathcal{C}[0]}$;
forall $(s, o) \in \mathcal{C}$ **do**
  if $s - \hat{o} > d$ **then**
    $\hat{C} \leftarrow \hat{C} \cup \{(s, o)\}$;
  else if *only_i is True and* $a_{es} = 1$ **then**
    $\hat{C} \leftarrow \hat{C} \cup \{(s, o)\}$; `/* The first word is aligned to a source word with a B- label */`
  else
    $c^* \leftarrow$ last added to $\hat{C}$;
    $\hat{C} \leftarrow \hat{C} \setminus \{c^*\}$;
    $\hat{C} \leftarrow \hat{C} \cup \{(c^*_s, o)\}$; `/* Merge ranges */`
  end
  $\hat{o} \leftarrow o$;
end

This phenomenon occurs when there is a missing alignment for a middle word within the range of words in the target sentence. Annotation collisions happen when a word in the target sentence aligns with words from different source entities, each having distinct labels. Incorrect alignments refer to situations where a word in the target sentence is wrongly aligned with a word from a source entity.

The projection utilizing word-to-word alignments is carried out through a heuristic algorithm described by [GAR22]. This algorithm (Algorithm 2) addresses split annotations by merging word ranges separated by a maximum of d words that are not aligned to any word from the source entity. A detailed merging process is provided in Algorithm 1, which includes an additional parameter, `only_i`, indicating whether the first word of the right-side merging range can be aligned with the first word of the source entity that has a label *B-*.

Annotation collisions are managed by assigning the label to the longest continuous aligned range. In the formulation of the algorithm that we present, this property is generalized, allowing for either the omission of any limits or the selection of the top-k continuous aligned ranges by length.

Furthermore, the original algorithm is extended to address incorrect alignments by introducing an optional threshold thr for the length ratio between a source entity and the corresponding range of words in the target sentence. Since a one-to-one correspondence in the number of words is typically expected for languages with similar writing systems, a small ratio generally indicates the presence of a single incorrectly aligned word.

Algorithm 2 : Heuristic projection algorithm based on word-to-word alignments

Data : S – set of source entities, a_{kl} – word-to-word alignments, parameters $d, k \in \mathbb{N}$, only_i $\in \{True, False\}, thr \in [0, 1]$

Result : $l^{tgt} = (l_1^{tgt}), \ldots, l_m^{tgt}$ – labelling of the target sentence

forall $l_i^{tgt} \in l^{tgt}$ **do**
  $l_i^{tgt} \leftarrow$ O ;
end
forall $p^{src} \in S$ **do**
  `/* Extract continuous ranges of maximum length of target words that are aligned to any word of the source entity */`
  $C^* \leftarrow \left\{(s,o) \middle| \forall r \in \{s, \ldots, o\}\ \exists i \in \{i_{p^{src}}, \ldots, j_{p^{src}}\}\ a_{ir} = 1\right\}$;
  $C \leftarrow \left\{(s,o) \in C^* \middle| \nexists (\hat{s}, \hat{o}) \in C^*\ [s,o] \subset [\hat{s}, \hat{o}]\right\}$;
  `/* Merge ranges that are separated by` d `non-aligned words */`
  $C \leftarrow$ merge$(C, a_{kl}, i_{p^{src}}, d,$ only_i$)$;
  `/* Length ratio thresholding (optional) */`
  if $thr > 0$ **then**
    $C \leftarrow \left\{(s,o) \in C \middle| \frac{o-s}{j_{p^{src}} - i_{p^{src}}} > thr\right\}$;
  end
  `/* Take only top-k longest aligned ranges (optional) */`
  if $k > 0$ **then**
    $C \leftarrow$ sort C by length and take top k ;
  end
  `/* Labelling */`
  forall $(s,o) \in C$ **do**
    `/* Label the range only if any word of it has not been previously labeled. */`
    if $\forall r \in \{s, \ldots, o\} \quad l_r = O$ **then**
      $\hat{l} \leftarrow l_{p^{src}}$; `/* Class of the source entity */`
      $l_s \leftarrow$ B-$\hat{l}$; `/* Assign the B- label to the first word */`
      forall $r \in \{s+1, o\}$ **do**
        $l_r \leftarrow$ I-$\hat{l}$; `/* Assign the I- label to consecutive words */`
      end
    end
  end
end

3 Methodology

The overall idea of formalizing the projection step of the XLNER pipeline is to formulate it as a matching process between source entities and ranges of words in the target sentence, referred to as target candidates. This process is illustrated in Figure 3.1. Whereas a predetermined set of source entities for projection is provided; a subset of continuous word ranges from the original sentence in the target language can be selected and considered as potential projections for source entities. The likelihood scores that each source entity should be projected onto each candidate can also be computed. The goal is to identify a combination of source entities and target candidates that maximizes the overall sum of the corresponding scores. However, natural constraints must also be taken into account. For instance, when two source entities project onto overlapping candidates, it becomes ambiguous to which source entity the overlapping words pertain. Therefore, projections onto overlapping candidates should be prohibited. Furthermore, it is logical to restrict the number of candidates to which each source entity can be projected, as a one-to-one correspondence between source entities and entities in the target sentence is typically anticipated. Based on these fundamental properties, we can establish an integer linear optimization problem.

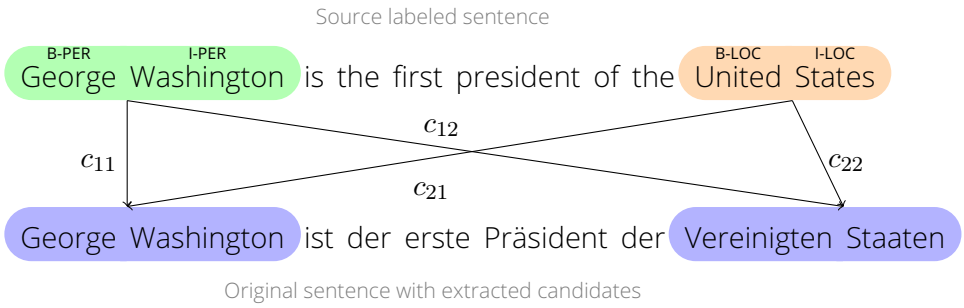


Figure 3.1: Illustration of the proposed idea of matching source entities and candidates in the target sentence

3.1 Formulation of the ILP problem

First and foremost, definitions of target candidate and overlapping relation should be established.

Definition 3.1 (Target candidate)**.** Let $s^{tgt} = \left(s_1^{tgt}, \ldots, s_m^{tgt}\right)$ be a target sentence consisting of m words. The target candidate $p^{tgt} = (i_{p^{tgt}}, j_{p^{tgt}})$ is defined as a continuous subrange of words within the sentence, characterized by the index $i_{p^{tgt}} \in \{1, \ldots, m\}$ of the first word and the index $j_{p^{tgt}} \in \{1, \ldots, m\}$ of the last word.

Definition 3.2 (Relation of overlapping)**.** Target candidates $p_1^{tgt} = (i_{p_1^{tgt}}, j_{p_1^{tgt}})$ and $p_2^{tgt} = (i_{p_2^{tgt}}, j_{p_2^{tgt}}) \in T$ are considered to overlap if and only if $(i_{p_1^{tgt}} \leq j_{p_2^{tgt}}) \wedge (i_{p_2^{tgt}} \leq j_{p_1^{tgt}})$, which indicates that the sets of word indices are not disjoint. Overlapping candidates will be denoted using the following notation: $p_1^{tgt} \cap p_2^{tgt} \neq \emptyset$.

Having defined all necessary objects, the ILP problem for the projection step of the XLNER pipeline that aligns with our requirements can now be formulated. Let S be a set of source entities, T – set of target candidates and $\cap \subset T^2$ is a relation of overlapping, then the projection ILP problem is the following:

$$\max_x \sum_{(p^{src}, p^{tgt}) \in S \times T} c_{p^{src}, p^{tgt}} x_{p^{src}, p^{tgt}} \tag{3.1}$$

subject to

$$\sum_{p^{tgt} \in T} x_{p^{src}, p^{tgt}} \lessgtr n_{proj} \qquad \forall p^{src} \in S \tag{3.2}$$

$$x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} \leq 1 \qquad \forall (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S, T) \tag{3.3}$$

$$x_{p^{src}, p^{tgt}} \in \{0, 1\} \qquad \forall (p^{src}, p^{tgt}) \in S \times T \tag{3.4}$$

where $n_{proj} \in \mathbb{N}$ and $\hat{\Pi}(S, T)$ represents a set of combinations of source entities and target candidates that cannot be projected onto together due to overlapping. This set is defined as follows:

$$\hat{\Pi}(S, T) = \left\{ (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \middle| p_1^{src}, p_2^{src} \in S, p_1^{tgt}, p_2^{tgt} \in T, \quad p_1^{tgt} \cap p_2^{tgt} \neq \emptyset, \right. \\ \left. (p_1^{src} \neq p_2^{src}) \vee (p_1^{tgt} \neq p_2^{tgt}) \right\} \tag{3.5}$$

Here, each variable $x_{p^{src}, p^{tgt}}$ indicates whether the source entity $p^{src} \in S$ is projected onto the target candidates $p^{tgt} \in T$. The set of constraints (3.3) ensures that it is impossible for one or more source entities to be projected onto overlapping target candidates. Additionally, another set of constraints (3.2) limits the number of projections for each source entity. It should be noted that the inequality sign is not fixed, as various forms may be relevant: "less," "less or equal," and "equal" can be used to ensure that a source entity is not projected onto as many available candidates as possible solely to increase the objective value. Conversely, "greater" or "greater or equal" can be employed to enforce that a source entity will be projected at least a specified number of times, e.g., once, ensuring that the solver does not simply ignore it. It is evident that $n_{proj} = 0$ represents a corner case that typically lacks meaning, except in the case of a "greater or equal" inequality, which effectively eliminates any limits and is equivalent to having no constraints of this type.

However, the number of constraints (3.3) scales quadratically on the the number of source entities. This considerable growth in constraints can complicate the problem-solving process.

Consequently, it is essential to explore methods to reduce the number of such constraints. To begin, let us examine the overlapping relation that forms the basis of this issue.

Lemma 3.1. *The relation of overlapping is not transitive.*

Proof. This will be demonstrated by providing a counterexample. Assume the following target candidates:

$$a = (2, 4) \quad b = (1, 2) \quad c = (4, 5)$$

By the definition of the overlapping relation $a \cap b \neq \emptyset$ and $a \cap c \neq \emptyset$ but b and c are not overlapping. □

This fact leads to the conclusion that it is impossible to partition all target candidates into groups of mutually overlapping candidates and therefore select at most one candidate from each group to match with a source entity.

Nevertheless, it is possible to reduce the number of constraints (3.3). The underlying idea for this reduction is that if it is impossible to project one or any two source entities onto overlapping candidates, then it is also impossible to project any number of source entities onto these candidates. Consequently, it suffices to sum the constraints across all source entities. This approach is feasible because all variables $x_{p^{src},p^{tgt}}$ are binary, and therefore, they cannot be negative.

Theorem 3.2. *The set of constraints* (3.3) *is satisfied if and only if the following sets of reduced constraints are satisfied:*

$$\sum_{p^{src} \in S} (x_{p^{src},p_1^{tgt}} + x_{p^{src},p_2^{tgt}}) \leq 1 \qquad \forall (p_1^{tgt}, p_2^{tgt}) \in \Pi(T)$$

$$\sum_{p^{src} \in S} x_{p^{src},p^{tgt}} \leq 1 \qquad \forall p^{tgt} \in T \,\Big|\, \nexists p_2^{tgt} \in T : p^{tgt} \neq p_2^{tgt}, p^{tgt} \cap p_2^{tgt} \neq \emptyset$$

where

$$\Pi(T) = \left\{ (p_1^{tgt}, p_2^{tgt}) \,\middle|\, p_1^{tgt}, p_2^{tgt} \in T, \quad p_1^{tgt} \cap p_2^{tgt} \neq \emptyset, p_1^{tgt} \neq p_2^{tgt} \right\}$$

Proof. Necessity: Assume that constraints (3.3) are satisfied, but the proposed constraints are not. Then there are two options:

$$\exists (p_1^{tgt}, p_2^{tgt}) \in \Pi(T) \,\Big|\, \sum_{p^{src} \in S} (x_{p^{src},p_1^{tgt}} + x_{p^{src},p_2^{tgt}}) > 1$$

or

$$\exists p^{tgt} \in T \,\Big|\, \nexists p_2^{tgt} \in T : p^{tgt} \neq p_2^{tgt}, p^{tgt} \cap p_2^{tgt} \neq \emptyset \,\Big|\, \sum_{p^{src} \in S} x_{p^{src},p^{tgt}} > 1$$

Consider the first option, then

$$\exists p_1^{src}, p_2^{src} \in S \Big| x_{p_1^{src}, p_1^{tgt}} = 1, x_{p_2^{src}, p_2^{tgt}} = 1 \implies x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} = 2 > 1$$

Or

$$\exists p_1^{src}, p_2^{src} \in S,\ p_1^{src} \neq p_2^{src},\ p^{tgt} \in \{p_1^{tgt}, p_2^{tgt}\} \Big| x_{p_1^{src}, p^{tgt}} = 1, x_{p_2^{src}, p^{tgt}} = 1 \implies$$
$$x_{p_1^{src}, p^{tgt}} + x_{p_2^{src}, p^{tgt}} = 2 > 1$$

This situation contradicts our initial assumption that the constraints (3.3) are satisfied.

The same result we can obtain for the second case:

$$\sum_{p^{src} \in S} x_{p^{src}, p^{tgt}} > 1 \overset{(3.4)}{\implies}$$
$$\exists p_1^{src}, p_2^{src} \in S,\ p_1^{src} \neq p_2^{src} \Big| x_{p_1^{src}, p^{tgt}} = 1, x_{p_2^{src}, p^{tgt}} = 1 \implies$$
$$x_{p_1^{src}, p^{tgt}} + x_{p_2^{src}, p^{tgt}} = 2 > 1$$

Thus, the necessity has been established.

Sufficiency: Suppose that the following constraints are satisfied:

$$\sum_{p^{src} \in S} (x_{p^{src}, p_1^{tgt}} + x_{p^{src}, p_2^{tgt}}) \leq 1 \qquad \forall (p_1^{tgt}, p_2^{tgt}) \in \Pi(T)$$

Therefore, the sum can take on values of 0 or 1. If the sum is zero, then no entities are projected onto any of the overlapping candidates, and consequently, the constraints (3.3) are not violated. Now, let us consider the case when the sum is equal to 1.

$$\sum_{p^{src} \in S} (x_{p^{src}, p_1^{tgt}} + x_{p^{src}, p_2^{tgt}}) = 1 \overset{(3.4)}{\implies}$$
$$\exists! p^{src} \in S,\ \exists! p^{tgt} \in \{p_1^{tgt}, p_2^{tgt}\} \Big| x_{p^{src}, p^{tgt}} = 1 \implies$$
$$\forall p_1^{src}, p_2^{src} \in S,\ x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} \leq 1$$

Therefore we proved that

$$x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} \leq 1 \qquad \forall (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}'(S, T)$$

where

$$\hat{\Pi}'(S,T) = \Big\{ (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \Big| p_1^{src}, p_2^{src} \in S, p_1^{tgt},\ p_2^{tgt} \in T, \quad p_1^{tgt} \cap p_2^{tgt} \neq \emptyset, \\ p_1^{tgt} \neq p_2^{tgt} \Big\}$$

But it is not the same as set (3.5):

$$\hat{\Pi}(S,T)\backslash\hat{\Pi}'(S,T) = \left\{(p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \middle| p_1^{src}, p_2^{src} \in S,\ p_1^{tgt}, p_2^{tgt} \in T,\ p_1^{tgt} = p_2^{tgt},\ p_1^{src} \neq p_2^{src}\right\}$$

So we need to check whether constraints (3.3) are satisfied on this set difference.

First of all, let's notice that

$$\forall (p_1^{tgt}, p_2^{tgt}) \in \Pi(T) \qquad \sum_{p^{src}\in S} (x_{p^{src},p_1^{tgt}} + x_{p^{src},p_2^{tgt}}) = \sum_{p^{src}\in S} x_{p^{src},p_1^{tgt}} + \sum_{p^{src}\in S} x_{p^{src},p_2^{tgt}} \leq 1 \implies$$

$$\forall p^{tgt} \in T \middle| \exists p_2^{tgt} \in T : p^{tgt} \neq p_2^{tgt}, p^{tgt} \cap p_2^{tgt} \neq \emptyset \qquad \sum_{p^{src}\in S} x_{p^{src},p^{tgt}} \leq 1$$

Therefore there exists at most one source entity $p^{src} \in S$ that are projected to the target candidate p^{tgt}. And using exactly the same derivation as above we get:

$$\forall p^{tgt} \in T \middle| \exists p_2^{tgt} \in T : p^{tgt} \neq p_2^{tgt}, p^{tgt} \cap p_2^{tgt} \neq \emptyset \qquad \sum_{p^{src}\in S} x_{p^{src},p^{tgt}} \leq 1 \overset{(3.4)}{\implies}$$

$$\sum_{p^{src}\in S} x_{p^{src},p^{tgt}} = 0 \overset{(3.4)}{\implies} \forall p^{src} \in S, x_{p^{src},p^{tgt}} = 0$$

$$\text{or} \qquad \implies$$

$$\sum_{p^{src}\in S} x_{p^{src},p^{tgt}} = 1 \overset{(3.4)}{\implies} \exists! p^{src} \in S, x_{p^{src},p^{tgt}} = 1$$

$$\forall p_1^{src}, p_1^{src} \in S,\ p_1^{src} \neq p_2^{src}$$

$$x_{p_1^{src},p^{tgt}} + x_{p_2^{src},p^{tgt}} \leq 1 \qquad \forall p^{tgt} \in T \middle| \exists p_2^{tgt} \in T : p^{tgt} \neq p_2^{tgt}, p^{tgt} \cap p_2^{tgt} \neq \emptyset$$

So, only the case, when a target candidate has no distinct overlapped target candidate remains. But this case is fully covered by the second part of the proposed reduced constraints:

$$\sum_{p^{src}\in S} x_{p^{src},p^{tgt}} \leq 1 \qquad \forall p^{tgt} \in T \middle| \nexists p_2^{tgt} \in T : p^{tgt} \neq p_2^{tgt}, p^{tgt} \cap p_2^{tgt} \neq \emptyset$$

And by considering that this sum can be either 0 or 1 and making exactly the same derivation we conclude that it implies that constraints (3.3) are satisfied. □

The theorem allows to reduce number of constraints, ensuring that they scale as a constant with respect to the number of source entities. Consequently, we obtain the following final

formulation of the ILP problem for the projection step of the XLNER pipeline.

$$
\begin{aligned}
&\max_{x} \sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} \\
&\text{subject to} \\
&\sum_{p^{tgt}\in T} x_{p^{src},p^{tgt}} \leqslant n_{proj} && \forall p^{src} \in S \\
&\sum_{p^{src}\in S} (x_{p^{src},p_1^{tgt}} + x_{p^{src},p_2^{tgt}}) \leq 1 && \forall (p_1^{tgt}, p_2^{tgt}) \in \Pi(T) \\
&\sum_{p^{src}\in S} x_{p^{src},p^{tgt}} \leq 1 && \forall p^{tgt} \in T \,\middle|\, \nexists p_2^{tgt} \in T : p^{tgt} \neq p_2^{tgt}, p^{tgt} \cap p_2^{tgt} \neq \emptyset \\
&x_{p^{src},p^{tgt}} \in \{0,1\} && \forall (p^{src}, p^{tgt}) \in S \times T
\end{aligned}
\tag{3.6}
$$

3.2 Candidates extraction

Whereas the set S of source entities is given, as all source entities have been extracted in the previous step of the XLNER pipeline, the construction of the set T of target candidates remains an open question.

The simplest method for candidate extraction is to consider all possible n-grams (continuous ranges of words) from the target sentence as candidates. This approach guarantees that no actual target entity will be excluded from the set of target candidates. However, from a computational perspective, this can pose a challenge, as the number of all n-grams scales quadratically. Furthermore, from an application standpoint, the majority of candidates generated in this manner are unlikely to represent valid target entities.

One of the simplest strategies to address this problem is to limit the maximum possible length of the candidates. From the perspective of Named Entity Recognition, it is reasonable to expect that actual target entities typically do not exceed a certain predefined length. For instance, it is unlikely for a person's name to consist of more than 10 words, and if there are three source entities, it is unlikely that there exists a single target entity that consists of all words of the target sentence. Thus, we obtain the algorithm 3 for candidates extraction wilt a bounded maximum length of n-grams.

Algorithm 3 : Bounded length n-gram candidates extraction

Data : $n \in \mathbb{N}$ – number of words in the target sentence, $M \leq n \in \mathbb{N}$ – maximum length of a target candidate
Result : T – set of target candidates

```
T ← ∅ ;
for i ← 0 to n do
    m ← min(i + M, n) ;
    for j ← i to m do
        T ← T ∪ {(i, j)} ;
    end
end
```

An alternative approach for candidate extraction is to employ a model to generate candidates. In this regard, TProjection [GAR23] utilizes a fine-tuned T5 model with beam search for candidate generation. In theory, any large language model can be fine-tuned for this purpose. However, the primary drawback of this approach is the necessity to fine-tune these models, which requires the availability of a labeled dataset in the target language or reliance on cross-lingual model transfer. Instead, in this context, it may be more feasible and effective to train a model to predict target entities directly rather than to generate candidates for the projection step of the XLNER pipeline. Moreover, autoregressive models tend to be resource-intensive, making it inefficient to utilize them solely for candidate extraction.

Alternatively, for candidate extraction a multilingual encoder-only Transformer [Vas17] model with high recall can be considered. However, these models also require training and therefore labelled dataset to extract candidates effectively. In the following chapters, these alternatives will not be discussed and will remain as topics for future work.

3.3 Matching scores

Thus far, the ILP problem (3.6) has been formulated and discussed, which involves projecting source entities onto target candidates using a matching score $c_{p^{src},p^{tgt}}$. This score represents the likelihood that a given source entity $p^{src} \in S$ should be projected onto the corresponding target candidate $p^{tgt} \in T$. However, the question of how to compute all these scores remains unresolved. In this section, various options for evaluating these scores will be proposed.

3.3.1 Alignment-based score

In Chapter 2, it was demonstrated that word-to-word alignments can be utilized for the projection step of the XLNER pipeline. However this was achieved through a heuristic algorithm. Nonetheless, an attempt can be made to incorporate these alignments into the proposed ILP problem by calculating matching scores using word-to-word alignments.

The matching score that inspired from this idea is the following:

$$c^{align}_{p^{src},p^{tgt}} = \frac{\sum_{k=i_{p^{src}}}^{j_{p^{src}}} \sum_{l=i_{p^{tgt}}}^{j_{p^{tgt}}} a_{kl}}{(j_{p^{src}} - i_{p^{src}}) + (j_{p^{tgt}} - i_{p^{tgt}})} \tag{3.7}$$

The form of the score is grounded in the natural properties expected from a score based on word-to-word alignments. First, the greater the number of aligned words between the source entity and the target candidates, the higher the score should be. Second, it is insufficient to simply divide the number of aligned words by the length of the source entity, as this would lead to a higher score for longer target candidates. Lastly, the third property, which is particularly beneficial: when considering two candidates, where one is a substring of the other, if the total number of aligned words between the source entity and these candidates is the same, preference should be given to the smaller candidate. The score defined in equation (3.7) fulfills this requirement.

Lemma 3.3. *Suppose that for a specific pair of source entity $p^{src} \in S$ and target candidate $p^{tgt} \in T$, the score defined in equation* (3.7) *is equal to c. If there exists an extended candidate $\hat{p}^{tgt} \in T$ that is increased by one word to the left or right, such that this additional word is not aligned with any word of the source entity, then the score* (3.7) *between the source entity and this target candidate will be lower than c.*

Proof. Considering the extension to the right, let $\hat{p}^{tgt} = (i_{p^{tgt}}, j_{p^{tgt}} + 1)$. Given that this additional word is not aligned with any word of the source entity, it implies that:

$$\sum_{k=i_{p^{src}}}^{j_{p^{src}}} a_{k,j_{p^{tgt}}+1} = 0$$

Then:

$$c^{align}_{p^{src},\hat{p}^{tgt}} = \frac{\sum_{k=i_{p^{src}}}^{j_{p^{src}}} \sum_{l=i_{p^{tgt}}}^{j_{p^{tgt}}+1} a_{kl}}{(j_{p^{src}} - i_{p^{src}}) + (j_{\hat{p}^{tgt}} - i_{\hat{p}^{tgt}})} =$$

$$\frac{\sum_{k=i_{p^{src}}}^{j_{p^{src}}} \sum_{l=i_{p^{tgt}}}^{j_{p^{tgt}}} a_{kl}}{(j_{p^{src}} - i_{p^{src}}) + (j_{p^{tgt}} + 1 - i_{p^{tgt}})} + \frac{\sum_{k=i_{p^{src}}}^{j_{p^{src}}} a_{k,j_{p^{tgt}}+1}}{(j_{p^{src}} - i_{p^{src}}) + (j_{p^{tgt}} + 1 - i_{p^{tgt}})} =$$

$$\frac{\sum_{k=i_{p^{src}}}^{j_{p^{src}}} \sum_{l=i_{p^{tgt}}}^{j_{p^{tgt}}} a_{kl}}{(j_{p^{src}} - i_{p^{src}}) + (j_{p^{tgt}} - i_{p^{tgt}}) + 1} < \frac{\sum_{k=i_{p^{src}}}^{j_{p^{src}}} \sum_{l=i_{p^{tgt}}}^{j_{p^{tgt}}} a_{kl}}{(j_{p^{src}} - i_{p^{src}}) + (j_{p^{tgt}} - i_{p^{tgt}})} = c$$

The derivation for the extension to the left, where $\hat{p}^{tgt} = (i_{p^{tgt}} - 1, j_{p^{tgt}})$, is structurally analogous. Thus, the lemma has been proven. □

Corollary 3.3.1. *The score of the extended candidate p^{tgt}_{+n} derived from the target candidate p^{tgt}, when extended to the left or right by $n \in \mathbb{N}$ non-aligned words that do not correspond to any word of the source entity $p^{src} \in S$, is lower than the score of the original candidate.*

Proof. Let us prove this statement using mathematical induction for the case of extension to the right (the proof for the left extension will follow a similar structure).

Base: By the lemma 3.3 $c^{align}_{p^{src},p^{tgt}} > c^{align}_{p^{src},p^{tgt}_{+1}}$.

Induction step: Assume that for some $k \in \mathbb{N}$ holds $c^{align}_{p^{src},p^{tgt}} > c^{align}_{p^{src},p^{tgt}_{+k}}$. Then by the lemma 3.3 $c^{align}_{p^{src},p^{tgt}_{+k}} > c^{align}_{p^{src},p^{tgt}_{+(k+1)}}$ and therefore $c^{align}_{p^{src},p^{tgt}} > c^{align}_{p^{src},p^{tgt}_{+(k+1)}}$.

Thus, by the mathematical induction, $c^{align}_{p^{src},p^{tgt}} > c^{align}_{p^{src},p^{tgt}_{+n}}$ □

The main advantage of this property is that it allows for the reduction of the set of target candidates without compromising optimal solutions. Specifically, it permits the consideration of candidates only from the subrange between the leftmost and rightmost words aligned to any word of any source entity.

Theorem 3.4. *Let $m, M \in \mathbb{N}$ denote the indices of the leftmost and rightmost words aligned to any words of any source entity. Then there exists an optimal solution for the problem* (3.6) *where the constraints* (3.2) *take the form of $<$ or $\leq$, such that all source entity are projected onto target candidates from the set T generated by the algorithm 3 ($n = M - m$), where all word indices are contained within the range $[m, M]$.*

Proof. Suppose that there is no optimal solution that satisfies this requirement. This implies the existence of a source entity $p^{src} \in S$ and a target candidate $\hat{p}^{tgt} \in T$ such that $(i_{\hat{p}^{tgt}} < m) \vee (M < j_{\hat{p}^{tgt}})$ and $x_{p^{src},\hat{p}^{tgt}} = 1$ for an optimal solution x.

This case can be divided into two parts: one where $\hat{p}^{tgt}$ contains no words aligned to any words of source entities, and the other where it does have such alignments.

Considering the first scenario, if $\hat{p}^{tgt}$ consists entirely of words that are not aligned to any word from source entities, then the alignment-based matching score, according to the definition in (3.7), equals 0.

In this situation, we can take a solution x^* that is identical to the optimal solution x, except for the target candidate $\hat{p}^{tgt}$; that is, we set $x_{p^{src},\hat{p}^{tgt}} = 0$. Consequently, the objective function remains unchanged because the corresponding matching score is zero. Additionally, we do not violate the non-overlapping constraints (3.3):

$$\begin{aligned}
&\forall p_1^{src} \in S, p_1^{tgt} \in T \Big| (p_1^{src}, p^{src}, p_1^{tgt}, \hat{p}^{tgt}) \in \hat{\Pi}(S, T) \\
&x_{p_1^{src},p_1^{tgt}} + x_{p^{src},\hat{p}^{tgt}} = x_{p_1^{src},p_1^{tgt}} + 1 \leq 1 \implies \\
&x^*_{p_1^{src},p_1^{tgt}} + x^*_{p^{src},\hat{p}^{tgt}} = x_{p_1^{src},p_1^{tgt}} + 0 \leq 1
\end{aligned}$$

As well as constraints (3.2):

$$\sum_{t \in T} x_{p^{src},t} \leq n_{proj} \implies$$

$$\sum_{t \in T} x^*_{p^{src},t} = \sum_{t \in T \setminus \{p^{tgt}\}} x_{p^{src},t} + x^*_{p^{src},p^{tgt}} = \sum_{t \in T \setminus \{p^{tgt}\}} x_{p^{src},t} + 0 \leq n_{proj}$$

Therefore, such a solution x^* is also optimal, just like the original solution x, but with target candidates constrained to have indices within the range $[m, M]$.

Considering the second option, where a target candidate $\hat{p}^{tgt}$ contains words that are aligned to words from the source entity p^{src}. Therefore the indices of its words overlap with the range $[m, M]$. By corollary 3.3.1, the candidate $p^{tgt} = (\max(i_{\hat{p}^{tgt}}, m), \min(M, j_{\hat{p}^{tgt}}))$ which is a substring of $\hat{p}^{tgt}$ restricted to the range $[m, M]$, will have a higher alignment-based matching score.

Let us consider a solution x^* that is identical to the solution x everywhere except for the target candidates $\hat{p}^{tgt}$ and p^{tgt}; specifically, $x^*_{p^{src},\hat{p}^{tgt}} = 0$ and $x^*_{p^{src},p^{tgt}} = 1$. This solution also satisfies all constraints of the ILP problem.

For the constraints (3.2), the proof is as follows:

$$\sum_{t \in T} x_{p^{src},t} = \sum_{t \in T \setminus \{p^{tgt}, \hat{p}^{tgt}\}} x_{p^{src},t} + x_{p^{src},p^{tgt}} + x_{p^{src},\hat{p}^{tgt}} =$$
$$\sum_{t \in T \setminus \{p^{tgt}, \hat{p}^{tgt}\}} x_{p^{src},t} + 0 + 1 \leq n_{proj} \implies$$
$$\sum_{t \in T} x^*_{p^{src},t} = \sum_{t \in T \setminus \{p^{tgt}, \hat{p}^{tgt}\}} x_{p^{src},t} + x^*_{p^{src},p^{tgt}} + x^*_{p^{src},\hat{p}^{tgt}} =$$
$$\sum_{t \in T \setminus \{p^{tgt}, \hat{p}^{tgt}\}} x_{p^{src},t} + 1 + 0 \leq n_{proj}$$

And for non-overlapping constraints, since p^{tgt} is a substring of $\hat{p}^{tgt}$ by a construction, we have:

$$\forall t \in T \quad t \cap p^{tgt} \neq \emptyset \implies t \cap \hat{p}^{tgt} \neq \emptyset$$

It gets us that all constraints that should be hold for p^{tgt} should also be satisfied for $\hat{p}^{tgt}$:

$$\left\{(p_1^{src}, p_1^{tgt}) \middle| (p_1^{src}, p^{src}, p_1^{tgt}, p^{tgt}) \in \hat{\Pi}(S,T)\right\} \subset \left\{(p_1^{src}, p_1^{tgt}) \middle| (p_1^{src}, p^{src}, p_1^{tgt}, \hat{p}^{tgt}) \in \hat{\Pi}(S,T)\right\} \tag{3.8}$$

Suppose that it is not the case for the solution x^*:

$$\exists p_1^{src} \in S, p_1^{tgt} \in T \Big| (p_1^{src}, p^{src}, p_1^{tgt}, p^{tgt}) \in \hat{\Pi}(S,T)$$
$$x^*_{p_1^{src},p_1^{tgt}} + x^*_{p^{src},p^{tgt}} = x_{p_1^{src},p_1^{tgt}} + 1 > 1 \implies$$
$$x_{p_1^{src},p_1^{tgt}} = 1 \implies$$
$$x_{p_1^{src},p_1^{tgt}} + x_{p^{src},\hat{p}^{tgt}} = 1 + 1 = 2 > 1$$

However, it contradicts our assumption that the solution x was feasible since $(p_1^{src}, p^{src}, p_1^{tgt}, \hat{p}^{tgt}) \in \hat{\Pi}(S,T)$ due to (3.8). And therefore it proves that the constraints (3.3) are satisfied by the solution x^*.

At the same time, the solution x^* has a higher objective value, which implies that the original solution x was not optimal. This leads to a contradiction with our assumption that there is no optimal solution that satisfies the requirements of the theorem. □

It is important to note that the theorem applies only to the constraints (3.2) in the forms of $<$ or $\leq$. This is due to the fact that the ILP problem may lose feasibility if projections with zero scores—those that exist solely to satisfy these constraints and have no practical application—are removed.

In comparison to the heuristic algorithm that employs word-to-word alignments, the formulation of the optimization problem with matching scores (3.7) presents a significant advantage: it allows for the evaluation of the confidence associated with each specific projection, as a score is available for each one. In contrast, heuristics only provide a labeling of the target sentence. Additionally, the predictions made by this formulation and the heuristics are not always the same. For example, the heuristic algorithm merges all continuous ranges of aligned words that are separated by at most a fixed number of non-aligned words, whereas the ILP formulation will only do so under specific conditions.

It is also worth mentioning that, for the alignment-based score, there is no fixed upper bound, as the maximum value of the numerator is the product of the lengths of the source entity and the target candidate. This limitation may be addressed by replacing one of the summation operations in the numerator of equation (3.7) with a logical OR. Nevertheless, in our experiments, we will utilize the original form of the alignment-based matching score.

3.3.2 NER model-based score

Another approach to evaluate matching scores between source entities and target candidates is to utilize a multilingual NER model. The underlying concept is as follows: for each word in the target candidate, the model predicts a probability distribution over a set of classes that determines the likelihood that a word has a specific label. To compute the matching score indicating how likely a source entity classified as l should be projected onto a target candidate, we calculate the average probability of class l across all words of the candidate.

Let L denote a set of classes. Since a NER model outputs results in the IOB format, where the first word of a predicted entity with class $l \in L$ is labeled as *B-l*, and subsequent words in the entity are labeled as *I-l*, the model's output can be represented as a matrix $p_{m,o}$. In this representation, $m \in \mathbb{N}$ is the index of a word, and $o \in \{1, 2|L| + 1\}$ is the index of a label. We will assume that the label O, which indicates that a word does not belong to any class, has an index of $2|L| + 1$.

Next, we introduce mappings $B[l] : L \to \{1, \dots, 2|L|\}$ and $I[l] : L \to \{1, \dots, 2|L|\}$, which return the indices in the probability matrix of the B and I labels, respectively, for a class $l \in L$.

Therefore, the NER model-based matching score can be expressed as follows:

$$c^{ner}_{p^{src},p^{tgt}} = \alpha^{(j_{p^{tgt}} - i_{p^{tgt}})-1} \frac{p_{i_{p^{tgt}},B[l_{p^{src}}]} + \sum\limits_{k=i_{p^{tgt}}+1}^{j_{p^{tgt}}} p_{k,I[l_{p^{src}}]}}{j_{p^{tgt}} - i_{p^{tgt}}} \tag{3.9}$$

where $l_{p^{src}} \in L$ denotes the class of the source entity $p^{src} \in S$ and $\alpha > 1 \in \mathbb{R}$ is a length-scaling constant.

The factor α is necessary to align the matching score with the NER model predictions: suppose two target candidates where the first one is a substring of the second and the second is identical to the predicted by a NER model entity. Then the matching cost of the first candidate should be lower than the matching cost of the second. But without setting $a > 1$ it is not always satisfied.

For example, let the set of classes contain only the class *PER*, so $L = \{\text{PER}\}$. Assume the set of source entities consists of only one entity p^{src} with the class *PER*. Let the set of target

candidates be $T = \{p_1^{tgt} = (1,1), p_2^{tgt} = (1,2)\}$, and the probability matrix predicted by the model is as follows:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} \textit{B-PER} & \textit{I-PER} & \textit{O} \end{array} \\ \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.2 & 0.8 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

By taking the maximum over the rows, it is revealed that the entity predicted by the model has the class *PER* and consists of words with indices 1 and 2. However, the NER model-based matching score, computed without the scaling factor α, yields the following results:

$$c^{ner}_{p^{src},p_1^{tgt}} = 0.9 \qquad c^{ner}_{p^{src},p_2^{tgt}} = 0.85$$

In this case, the ILP problem will preferentially project the source entity onto a substring of the entity predicted by the model. Therefore, the length-scaling constant α cannot be omitted.

If α is a crucial component of the score, then it is worthwhile to evaluate the range of its possible values. To this end, let us notice the following observation.

Lemma 3.5. *Let $p_i, i \in K$ be a probability distribution over a finite set K with cardinality k. Then*

$$\max_{i \in K} p_i \geq \frac{1}{k}$$

Proof. Suppose that it is wrong, i.e.

$$\max_{i \in K} p_i < \frac{1}{k}$$

Then the sum of all probabilities:

$$\sum_{i \in K} p_i \leq \sum_{i \in K} \max_{j \in K} p_j = k \cdot \max_{j \in K} p_j < 1$$

However, since the sum of all probabilities in the distribution must equal 1, our assumption was incorrect, which proves the lemma. □

Having this useful fact it is possible to provide some estimates for the constant α.

Theorem 3.6. *Consider a source entity $p^{src} \in S$ and two target candidates $p_1^{tgt} = (i, j+1)$ and $p_2^{tgt} = (i, j)$. Suppose the predictions p from a NER model satisfy the following constraints:*

$$B[l_{p^{src}}] = \underset{l \in \{1,\ldots,2|L|+1\}}{\arg\max} p_{i,l} \qquad I[l_{p^{src}}] = \underset{l \in \{1,\ldots,2|L|+1\}}{\arg\max} p_{k,l} \; \forall k \in \{i+1, \ldots, j+1\}$$

i.e all these candidates are substrings of the entity predicted by the model. Let $M = \frac{1}{2|L|+1}$ and $n = j - i$. Then:

$$\alpha > 1 + \frac{1-M}{1+M} \implies c^{ner}_{p^{src},p_2^{tgt}} > c^{ner}_{p^{src},p_1^{tgt}}$$

Proof. Starting with the desired inequality we have:

$$
\begin{aligned}
&c^{ner}_{p^{src},p_2^{tgt}} > c^{ner}_{p^{src},p_1^{tgt}} \Leftrightarrow \\
&\alpha^n \frac{p_{i,B[l_{p^{src}}]} + \sum\limits_{k=i+1}^{j+1} p_{k,I[l_{p^{src}}]}}{n+1} > \alpha^{n-1} \frac{p_{i,B[l_{p^{src}}]} + \sum\limits_{k=i+1}^{j} p_{k,I[l_{p^{src}}]}}{n} \Leftrightarrow \\
&\alpha > \frac{(n+1) \cdot \left(p_{i,B[l_{p^{src}}]} + \sum\limits_{k=i+1}^{j} p_{k,I[l_{p^{src}}]} \right)}{n \cdot \left(p_{i,B[l_{p^{src}}]} + \sum\limits_{k=i+1}^{j+1} p_{k,I[l_{p^{src}}]} \right)} = \frac{(n+1) \cdot \left(p_{i,B[l_{p^{src}}]} + \sum\limits_{k=i+1}^{j} p_{k,I[l_{p^{src}}]} \right)}{n \cdot \left(p_{i,B[l_{p^{src}}]} + \sum\limits_{k=i+1}^{j} p_{k,I[l_{p^{src}}]} + p_{j+1,I[l_{p^{src}}]} \right)}
\end{aligned}
$$

To simplify the derivation, we will introduce aliases for certain quantities:

$$
S = p_{i,B[l_{p^{src}}]} + \sum_{k=i+1}^{j} p_{k,I[l_{p^{src}}]} \qquad s = p_{j+1,I[l_{p^{src}}]}
$$

Note that, according to lemma 3.5, the following inequalities hold:

$$
M \leq s \leq 1 \qquad nM \leq S \leq n \tag{3.10}
$$

Then we have:

$$
\alpha > \frac{(n+1)S}{n(S+s)} = \frac{nS + S + ns - ns}{nS + ns} = 1 + \frac{S - ns}{n(S+s)}
$$

The derivatives of this expression are the following:

$$
\left(1 + \frac{S-ns}{n(S+s)}\right)'_s = \frac{-n^2 S - nS}{n^2(S+s)^2} < 0 \qquad \left(1 + \frac{S-ns}{n(S+s)}\right)'_S = \frac{nS + n^2 s}{n^2(S+s)^2} > 0
$$

Hence, the maximum of the function is located at the point $s = M$ and $S = n$. Thus, we can select α such that:

$$
\alpha > 1 + \frac{1-M}{1+M} \overset{n \geq 1}{\geq} 1 + \frac{n - nM}{n(n+M)} \overset{\max}{\geq} 1 + \frac{S-ns}{n(S+s)}
$$

□

As demonstrated in the proof, this value represents just an upper estimate of such an α for the worst-case scenario. In practice, it may be reasonable to attempt even smaller values of α. However, doing so does not guarantee that the desired properties will be satisfied.

The primary distinction between a standard model transfer and the XLNER pipeline, which employs the ILP projection problem utilizing NER model-based matching scores, lies in the fact that in the latter, we need to have a projection from source entities to preserve predicted by a model target entities. Consequently, certain predictions made by a NER model, which may be incorrect, can be disregarded since the solution to the ILP problem may not include a projection from any source entity onto candidates corresponding to wrongly predicted target entities.

One of the challenges associated with neural network models is that they often exhibit overconfidence in their predictions, producing high probabilities for each output, including those that are incorrect. Since the proposed score directly relies on these probabilities, it can be adversely affected. To mitigate this issue, calibration [Guo+17] of the model can be implemented. The simplest method to achieve this is to scale the model's output logits by a temperature factor. Nevertheless, in the subsequent chapters, we will utilize NER models without any calibration.

The NER model-based matching score (3.9) possesses another noteworthy property. Since the computation of the score relies solely on the class of the source entity, the scores will be identical for all source entities sharing the same class. From the perspective of the ILP problem (3.6), this can result in a solution where a source entity is projected onto a semantically incorrect candidate; nonetheless, a correct class will still be assigned to the target candidate.

3.3.3 Translation-based score

Since every projected target entity should be an exact translation of its corresponding source entity, a matching score can be formulated based on the probability that a source entity translates into a target candidate.

The NMTScore [VS22] is a method for computing such a translation score given a neural machine translation model capable of translating between the desired languages.

The NMTScore proposes three translation-based measures: direct, pivot, and cross-likelihood. However, since the computation of the latter two measures requires significantly more computational resources than the first, the direct measure is chosen for our matching score. The direct NMTScore operates with the direct translation probability, which is computed as follows: let A and B be phrases in two languages a and b, and let p_θ represents the probabilities generated by a translation model. The direct translation probability is given by:

$$P_{direct}(A, B) = \left[\prod_{t=0}^{|A|} p_\theta(A^i | B, A^{<i}) \right]^{\frac{1}{|A|}}$$

Essentially, this is a probability, induced by an autoregressive translation model, normalized by the sequence length. The directed NMTScore similarity is a normalized version of it that constrains it to be less or equal to 1:

$$\widehat{sim}(A, B) = \frac{P_{direct}(A, B)}{P_{direct}(A, A)}$$

However, because the translation probability is a directed measure, it will differ if the arguments are swapped. To make the similarity score symmetric with respect to the arguments, the authors compute an average of both directions:

$$sim(A, B) = \frac{\widehat{sim}(A, B) + \widehat{sim}(B, A)}{2}$$

Thus, the translation-based matching score can be computed as the direct NMTScore of the given source entity and target candidate:

$$c^{nmt}_{p^{src},p^{tgt}} = sim\left(w^{src}[i_{p^{src}} : j_{p^{src}}], w^{tgt}[i_{p^{tgt}} : j_{p^{tgt}}]\right) \tag{3.11}$$

where w^{src}, w^{tgt} is an array of words of the source and target candidates respectively.

The quality of this type of matching score is heavily dependent on the performance of the translation model and its ability to accurately translate not just entire sentences, but also arbitrary subphrases.

3.3.4 Fused score

The performance of the alignment, NER, and translation models utilized for the computation of the scores discussed above varies based on factors such as language, domain, set of classes, etc. Each matching score has its own advantages and drawbacks; however, a natural way to minimize failures is to combine the scores. Since all scores are real numbers, one approach to achieve this is to compute a weighted sum of scores across all types:

$$c^{fused}_{p^{src},p^{tgt}} = \lambda_{align} c^{align}_{p^{src},p^{tgt}} + \lambda_{ner} c^{ner}_{p^{src},p^{tgt}} + \lambda_{nmt} c^{nmt}_{p^{src},p^{tgt}} \tag{3.12}$$

where $\lambda_{align} \geq 0, \lambda_{ner} \geq 0, \lambda_{nmt} \geq 0 \in \mathbb{R}$.

Furthermore, the fused score enhances the ability to address issues associated with the individual scores. For instance, the NER model used in the NER-based score (3.9) may occasionally correctly predict the spans of entities in the target sentence but misclassify their labels. During the computation of the NER model-based score, we assume that the labels are predicted accurately, as there is no method to amend them. However, when the score is comprised of various basic scores, we can utilize the NER model-based score solely to evaluate the likelihood that a given candidate can represent a target entity of any class, while determining the label through other types of scores. The modified version of (3.9) that implements this concept is as follows:

$$c^{ner}_{p^{src},p^{tgt}} = \alpha^{(j_{p^{tgt}} - i_{p^{tgt}})-1} \max_{l \in L} \frac{p_{i_{p^{tgt}},B[l]} + \sum_{k=i_{p^{tgt}}+1}^{j_{p^{tgt}}} p_{k,I[l]}}{j_{p^{tgt}} - i_{p^{tgt}}} \tag{3.13}$$

3.4 Analysis of the ILP problem

Although the proposed ILP problem is a specific instance of the well-studied general binary programming problem, it does not necessarily inherit all properties associated with it. For instance, while the maximum bipartite matching problem [PS03] can be formulated as an instance of the ILP problem, it can still be solved in polynomial time, whereas the general case of ILP is NP-hard. This distinction necessitates a more in-depth analysis of the proposed formulation.

3.4.1 Complexity

Constraints play a critical role in determining the complexity of the proposed problem. In the scenario where all candidates are non-overlapping and the constraints (3.2) take the form of equalities with $n_{proj} = 1$, the problem simplifies to an instance of the weighted bipartite matching problem [PS03], which belongs to complexity class P. However, let us explore the more general case.

For the sake of convenience, we will utilize the first formulation of the ILP problem (3.1)–(3.4) in this section. However, since it is equivalent to the formulation (3.6), this choice does not impact the results of the analysis.

A common approach to establishing the complexity of a problem is to reduce another problem with a known complexity to an instance of the problem under study. Therefore, let us consider the maximum weight independent set problem [PS03].

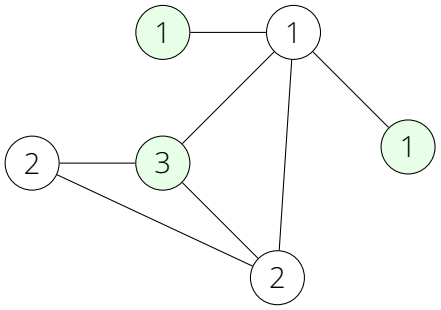


Figure 3.2: Maximum weigh independent set problem (MaxWIS). Vertices that form an optimal solution are colored in green

The maximum weight independent set problem (MaxWIS) involves finding a subset of vertices in an undirected graph with maximum total weight, such that no two vertices in the subset are connected by edges in the graph. An example of the maximum weight independent set problem is illustrated in Figure 3.2. The formal definition of MaxWIS as an ILP problem is provided in Definition 3.3.

Definition 3.3 (MaxWIS)**.** Let $G = (V, E, w), V \neq \emptyset,\ w : V \to \mathbb{R}$ be an undirected weighted graph, then a maximum weigh independent set problem for the graph G is the following:

$$\begin{aligned} &\max \sum_{v \in V} w_v x_v \\ &x_u + x_v \leq 1 \qquad \forall \{u, v\} \in E \\ &x_v \in \{0, 1\} \end{aligned}$$

In the scenario where all weights are equal to 1, the problem is referred to as the maximum independent set (MaxIS) problem. In this case, the objective is to maximize the cardinality of the independent set.

It can be demonstrated that the generalized form of the projection ILP problem (3.1)–(3.4), where the relation $\cap$ is an arbitrary reflexive, symmetric, and non-transitive relation, is NP-hard. The reduction of the maximum independent set problem to the instance of the generalized ILP problem that proves it is presented in Appendix B. However, the overlapping

relation defined in 3.2, which is employed in the projection ILP problem, represents a specific case in which the induced graph of the MaxIS problem is an interval graph. Consequently, the reduction from the proof is no longer valid. Moreover, it is known that for interval graphs, the maximum independent set problem can be solved in polynomial time [Bha+14].

Nevertheless, these results motivate further analysis. It turns out that the maximum weight independent set problem on interval graphs can also be solved in polynomial time [PB96]. So, we can prove that the non-overlapping constraints themselves do not make the entire projection ILP problem (3.1)–(3.4) computationally hard.

Consider the projection ILP problem without the constraints (3.2):

$$\begin{aligned}
&\max_{x} \sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} \\
&\text{subject to} \\
&x_{p_1^{src},p_1^{tgt}} + x_{p_2^{src},p_2^{tgt}} \leq 1 \qquad \forall (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S,T) \\
&x_{p^{src},p^{tgt}} \in \{0,1\} \qquad \forall (p^{src}, p^{tgt}) \in S \times T
\end{aligned} \tag{3.14}$$

It is possible to reduce this problem to an instance of the maximum weight independent set problem on an interval graph.

The idea of the reduction is straightforward. For every target candidate $t_v \in T$, we will create a distinct vertex $v \in V$ that corresponds to this candidate. The weight of each vertex v is computed based on the matching scores between all source entities and the target candidate that corresponds to this vertex:

$$w_v = \max_{s\in S} c_{s,t_v}$$

The conversion between the solutions of the ILP problem (3.14) and the solutions of the MaxIS problem can be established as follows:

$$\begin{aligned}
&s_v^* = \arg\max_{s\in S} x_{s,t_v} \\
&x_v = 1 \implies \begin{cases} x_{s^*,t_v} = 1 \\ x_{s,t_v} = 0 \quad \forall s \in S \setminus \{s^*\} \end{cases} \qquad \exists s \in S \Big| x_{s,t_v} = 1 \implies x_v = 1
\end{aligned} \tag{3.15}$$

That is, if the vertex $v \in V$ belongs to the maximum weight independent set then the source entity with the highest matching score is projected onto the target candidate p_v^{tgt} corresponding to this vertex. In the case when several source entities have equal maximal scores, we select only one entity.

In the instance of the MaxWIS problem, induced by the projection ILP problem, nodes $v, u \in V$, where $u \neq v$, are connected if and only if their corresponding target candidates overlap:

$$\{v,u\} \in E \Leftrightarrow t_u \cap t_v \neq \emptyset \tag{3.16}$$

Furthermore, since the set of target candidates is, by definition, a set of intervals, the corresponding graph will indeed be an interval graph. This completes the reduction procedure. It is important to note that the reduction is executed in polynomial time.

Given the one-to-one correspondence between the vertices in the MaxWIS problem and the target candidates T, we can consider the following set that is completely determined by and determines the solution of the MaxWIS problem:

$$T_x^* = \{t \in T | \exists s \in S, x_{s,t} = 1\}$$

It is important to note that all feasible solutions of the MaxWIS will correspond to feasible solutions of the ILP problem (3.14) and vice versa.

Lemma 3.7. *Suppose x_v is a feasible solution of the maximum weight independent set problem; then, there exists a corresponding feasible solution for the problem* (3.14)*.*

Proof. Suppose that a solution x that corresponds to a feasible solution x_v of the MaxWIS problem is non-feasible. Then, we have:

$$\exists (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S,T) \Big| x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} > 1$$

However, since the solution x_v is feasible, the constraints involving $p_1^{tgt} \neq p_2^{tgt}$ cannot be violated:

$$\begin{aligned}
&\forall \{u, w\} \in E \quad x_u + x_w \leq 1 \overset{(3.16)}{\implies} \\
&\forall p_1^{tgt} \in T_x^* \ \nexists p_2^{tgt} \in T_x^* \Big| p_1^{tgt} \cap p_2^{tgt} \neq \emptyset \implies \\
&\forall p_1^{src}, p_2^{src} \in S \quad x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} \leq 1
\end{aligned}$$

Therefore, $p_1^{tgt} = p_2^{tgt}$, and the constraints are violated because there exist at least two distinct source entities $p_1^{src}, p_2^{src} \in S$ that are projected onto the same target candidate $p^{tgt} \in T$:

$$x_{p_1^{src}, p^{tgt}} = 1 \qquad x_{p_2^{src}, p^{tgt}} = 1$$

However, this contradicts the fact that, by the construction of the corresponding solution to the ILP problem (3.15), there is only one source entity projected onto each target candidate. □

Lemma 3.8. *Suppose x is a feasible solution of the problem* (3.14)*; then, the corresponding solution to the maximum weight independent set problem is also feasible.*

Proof. By construction, the non-overlapping constraints of the problem (3.14) imply the constraints of the MaxWIS problem:

$$\begin{aligned}
&\forall (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S,T) \quad x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} \leq 1 \implies \\
&\forall p_1^{tgt}, p_2^{tgt} \in T, p_1^{tgt} \neq p_2^{tgt}, p_1^{tgt} \cap p_2^{tgt} \neq \emptyset \quad \nexists p_1^{src}, p_2^{src} \in S \Big| x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} > 1 \implies \\
&\nexists p_1^{tgt}, p_2^{tgt} \in T, p_1^{tgt} \neq p_2^{tgt}, p_1^{tgt} \cap p_2^{tgt} \neq \emptyset \Big| p_1^{tgt} \in T_x^*, p_2^{tgt} \in T_x^* \implies \\
&\forall \{v_{p_1^{tgt}}, v_{p_2^{tgt}}\} \in E \quad x_{v_{p_1^{tgt}}} + x_{v_{p_2^{tgt}}} \leq 1
\end{aligned}$$

□

Finally, we can demonstrate that it is possible to reduce the projection ILP problem, without the constraints (3.2), to an instance of the maximum weight independent set problem on an interval graph. Consequently, it can be solved using a polynomial-time algorithm.

Theorem 3.9. *The formulation* (3.14) *of the projection ILP problem, which does not include constraints on the number of target candidates projected from each source entity, can be solved in polynomial time.*

Proof. Let x denote the corresponding solution to the optimal solution x_v of the MaxWIS problem, which has been constructed through the reduction from the ILP problem. If x is always an optimal solution for the problem (3.14) then this would establish that we can reduce this problem to an instance of the maximum weighted independent set problem, which has a polynomial time algorithm, i.e. create induced MaxWIS problem, solve it with a polynomial time algorithm and convert solution of the MaxWIS problem back to the optimal solution of the original ILP problem.

By Lemma 3.7, the feasibility of the solution of the MaxWIS problem ensures the feasibility of the corresponding solution of the ILP problem. The objective function for the solution x is, by construction as indicated in (3.15), equal to the objective function of the MaxWIS problem:

$$\sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} = \sum_{v\in V} w_v x_v$$

Assume that there exists an optimal solution x^* of the problem (3.14) such that it is better than x:

$$\sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} < \sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x^*_{p^{src},p^{tgt}}$$

Consequently, for the corresponding solution x^*_v of the MaxWIS problem, we obtain:

$$\sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x^*_{p^{src},p^{tgt}} \leq \sum_{(p^{src},p^{tgt})\in S\times T} \max_{s\in S} c_{s,p^{tgt}} x^*_{p^{src},p^{tgt}} = \sum_{v\in V} w_v x^*_v$$

By Lemma 3.8, it follows that x^*_v will be a feasible solution. However, this contradicts the fact that x_v was an optimal solution of the MaxWIS problem:

$$\sum_{v\in V} w_v x_v = \sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} < \sum_{(p^{src},p^{tgt})\in S\times T} c_{p^{src},p^{tgt}} x^*_{p^{src},p^{tgt}} \leq \sum_{v\in V} w_v x^*_v$$

□

The question of whether the full projection ILP problem (3.6) is NP-hard still remains open. However, if it is NP-hard, the constraints (3.2) are principal factor of that complexity.

3.4.2 Approaches to compute the solution of the problem

Like any ILP problem, the problem (3.6) can be solved using methods such as branch-and-bound [LD10], cutting planes [GG63; Dyc81], branch-and-cut [PR91], and other exact algorithms. However, these approaches may require significant time to find an optimal

solution, which can make such formulations inefficient from an application standpoint, particularly when it is necessary to solve instances of this problem for thousands of sentences within a limited timeframe.

This situation highlights the need for an approximate algorithm capable of computing a solution that may not always be optimal but adheres to crucial constraints. One approach for developing such an algorithm for the problem (3.6) involves iteratively assigning 1 to the variable with the highest matching score and then removing all target candidates that overlap with the selected candidate to enforce non-overlapping constraints. Algorithm 4 represents a variant of this greedy algorithm.

Algorithm 4 : Approximate greedy algorithm for the proposed ILP problem

Data : instance of the ILP problem (3.6)
Result : x – "solution" of the ILP problem
$x \leftarrow 0$;
$P \leftarrow 0$; `/* number of projections by source entity */`
while $\exists p^{src} \in S, p^{tgt} \in T \Big| c_{p^{src},p^{tgt}} > 0$ **do**
  $s, t \leftarrow \underset{p^{src} \in S, p^{tgt} \in T}{\arg\max} \; c_{p^{src},p^{tgt}}$;
  $x_{s,t} \leftarrow 1$;
  $P_s \leftarrow P_s + 1$;
  forall $\hat{t} \in T \Big| \hat{t} \cap t \neq \emptyset$ **do** `// remove all overlapping with` t `candidates`
    $c_{s,\hat{t}} \leftarrow 0$;
  end
  `/* try to ensure constraints` (3.2) `*/`
  if *constraints* (3.2) *is a type of* $=, \leq$ **then**
    if $P_s = n_{proj}$ **then**
      forall $\hat{t} \in T$ **do**
        $c_{s,\hat{t}} \leftarrow 0$;
      end
    end
  end
  if *constraints* (3.2) *is a type of* $<$ **then**
    if $P_s = n_{proj} - 1$ **then**
      forall $\hat{t} \in T$ **do**
        $c_{s,\hat{t}} \leftarrow 0$;
      end
    end
  end
end

The steps of Algorithm 4 imply that by removing all overlapping target candidates—by setting their scores to zero—the algorithm maintains the non-overlapping constraints. Constraints (3.2) regarding the number of candidates projected from each source entity are ensured by removing all source entities that reach maximum allowed number of projections. However, this holds true only in cases where the constraints are of the form $<$ or $\leq$.

In the cases of $=, >$, or $\geq$, the algorithm may produce a "solution" that violates the constraint (3.2) under two specific circumstances. The first occurs when there is a source

entity for which all target candidates, which do not overlap with the already projected ones, initially have a zero matching score. However, in such cases, it does not make sense from an application standpoint to project the source entity onto these candidates, as they are definitively not counterparts of the source entity in the target sentence; otherwise, their matching score would not be zero.

The second situation arises when target candidates that could potentially be projected onto have been eliminated in previous iterations of the algorithm. But addressing this issue is challenging without backtracking or violating the feasibility of the non-overlapping constraints (3.3).

Nevertheless, Algorithm 4 operates in linear time with respect to the number of variables and, thus, can solve the ILP problem significantly faster than an exact ILP solver in general cases.

Even in cases where the output of the greedy algorithm is a feasible solution, it does not necessarily imply that the solution is optimal – there may exist another feasible solution with a higher objective value. This can be easily illustrated with an example.

Consider the ILP problem (3.6) with $n_{proj} = 2$ and the $\leq$ type of constraints (3.2). Let $T = \{p_1^{tgt} = (1, 2), p_2^{tgt} = (2, 3), p_3^{tgt} = (3, 5)\}$ and $S = \{p^{src}\}$, and let the matching scores be as follows:

$$c_{p^{src},p_1^{tgt}} = 0.2 \qquad c_{p^{src},p_2^{tgt}} = 0.3 \qquad c_{p^{src},p_3^{tgt}} = 0.2$$

The output of Algorithm 4 is the solution x with only one non-zero variable, $x_{p^{src},p_2^{tgt}} = 1$. This solution has an objective value of 0.3. In contrast, the actual optimal solution involves projecting the source entity onto both p_1^{tgt} and p_3^{tgt}, yielding an objective value of 0.4.

4 Evaluation

The evaluation strategies for the XLNER pipelines can be classified into two groups: extrinsic and intrinsic. Extrinsic evaluation involves utilizing the XLNER pipeline to produce a labeled dataset in the target language, followed by training a NER model on this dataset. In this context, the performance of the trained NER model serves as a measure of quality. A primary concern with this approach is its dependency on the training procedure; that is, results may vary due to different random sampling of batches, alterations in batch size, mixing of generated data with manually labeled data, other factors. Furthermore, this process demands substantial computational resources to train a model, which becomes problematic when conducting hundreds of experiments.

Consequently, the results of intrinsic evaluation will be presented in this chapter. This entails taking a manually labeled dataset in the target language, feeding it to the pipeline to generate labels, and comparing the predictions with the ground truth labels. It is reasonable to anticipate that superior intrinsic performance will correlate with improved extrinsic performance, assuming a consistent training setup.

To assess the proposed formulation of the projection step of the XLNER pipeline as an ILP problem (3.6) and to analyze various forms of it, two sets of experiments have been conducted. The first set evaluates the performance of the projection step in isolation. The second set of experiments tests the quality of the proposed method within the complete XLNER pipeline on the MasakhaNER2 dataset [Ade+22]. The source code utilized for conducting all experiments is publicly available on GitHub[1]. The GUROBI optimizer [Gur24], employed under an academic license, served as the exact ILP solver for all experiments.

Word-to-word alignments for all experiments were computed using a non-fine-tuned AWESOME [DN21] aligner with the following default hyperparameters: extraction method set to softmax, softmax_threshold of 0.001, and align_layer is 8.

For source labelling, candidate evaluation, and model transfer, the MDeBERTa-v3-base [HGC21] model, fine-tuned on the English split of the CONLL-2003 dataset [TD03], was utilized. This selection is based by the findings of the original study in which MasakhaNER2 was introduced, demonstrating that MDeBERTa-v3 achieves superior performance compared

[1] https://github.com/ShkalikovOleh/master-thesis

to other multilingual non-African-centric models, despite being smaller than XLM-RoBERTA-Large [Con+20]. The model was trained for 5 epochs, with a total batch size of 32 (consisting of 16 with gradient accumulation every 2 steps), utilizing the Adam optimizer [KB14] with betas set to $(0.9, 0.999)$ and a learning rate of $2 \cdot 10^{-5}$. The model is publicly accessible on the HuggingFace Hub[2].

In all experiments involving NER model-based and translation-based matching scores, in order to simplify the problem (by making scores matrix sparse) and based on the interpretation of the scores (where small values indicate a poor projection), all (3.9) scores were thresholded at 0.05 and all (3.11) scores were thresholded at 0.1. The α parameter of the NER model-based score was set to 1.15. The translation model utilized for the computation of the NMTScore, and consequently for the translation-based matching scores, is the distilled 600M version of the NLLB-200 [Tea+22].

All experiments involving target candidate extraction were conducted using the algorithm 3 with $n = 10$. Although it restricts the proposed method's ability to predict longer target entities, it remains within the 0.99 percentile of entity lengths for all datasets utilized. Furthermore, all experiments with only the alignment-based matching score were conducted with a reduced set of target candidates, as specified in Theorem 3.4.

4.1 Isolated evaluation of the projection step

The XLNER pipeline comprises three steps: forward translation, source NER labelling, and projection. Each of these steps can introduce errors. The new ILP-based approach for the projection step has been proposed, and it is essential to evaluate the performance of this step independently from errors associated with the preceding steps.

This necessitates the availability of a labelled dataset with parallel texts, which would enable the exclusion of the translation and source labelling phases. The Europarl-NER dataset [Age+18] serves as such a dataset. This dataset consists of 799 parallel sentences derived from the Europarl corpus [Koe05], and manually annotated according to four entity types, adhering to the CoNLL 2002 and 2003 guidelines for four languages: English, German, Italian, and Spanish.

Consequently, it becomes feasible to evaluate the performance of the proposed projection step as the ILP problem in isolation. Prior to this, however, it is important to investigate the performance of the heuristic word-to-word alignment-based algorithm 2 to determine the optimal combination of hyperparameters and the resulting performance metrics for comparison with the proposed projection step. The F1 scores for the heuristic algorithm, assessed with varying hyperparameters, are presented in Table 4.1.

The results indicate that the optimal combination of hyperparameters for the majority of languages is as follows: a merging distance $d = 1$, k should not be limited (an unrestricted maximum number of aligned subranges sorted by length that can be projected (i.e.,)), merging of aligned ranges should disregard whether the first word of the right-aligned range

[2] https://huggingface.co/ShkalikovOleh/mdeberta-v3-base-conll2003-en

d	k	only_i	tgt_lang thr	de	es	it
0	1	False	0.8	0.814	0.873	0.838
			-	0.896	0.819	0.789
		True	0.8	0.814	0.873	0.838
			-	0.896	0.819	0.789
	-	False	0.8	0.814	0.872	0.840
			-	0.875	0.763	0.735
		True	0.8	0.814	0.872	0.840
			-	0.875	0.763	0.735
1	1	False	0.8	0.819	**0.903**	0.871
			-	**0.916**	0.875	0.846
		True	0.8	0.815	0.886	0.848
			-	0.899	0.847	0.813
	-	False	0.8	0.819	**0.903**	**0.873**
			-	0.912	0.858	0.832
		True	0.8	0.815	0.885	0.850
			-	0.886	0.816	0.782

Table 4.1: Overall F1 scores for word-to-word alignments-based heuristic algorithm with different hyperparameter on the Europarl NER dataset

is aligned to the first word of a source entity, and a word length ratio threshold of 0.8. Hence, these hyperparameters will be employed for subsequent experiments.

The overall results are the following. The most significant improvement in performance occurs when the algorithm is permitted to merge aligned ranges together, provided that only one non-aligned word exists between them. This enhancement is attributed to the algorithm's ability to fill gaps in imperfect or missed alignments. Concurrently, imposing a limit k on the number of projected target ranges for any source entity results in a slight decrease in performance. Additionally, due to errors in alignments, the merging of aligned subranges, which only occurs when the right range begins with a word aligned to the first word of a source entity, leads to outcomes that are less favorable in comparison to scenarios without such a limit. Finally, applying a threshold on the length ratio between the source entity and the ranges of aligned target words enhances performance. This improvement is attributed to the filtering out wrong alignments of single words, which would otherwise be incorrectly part of projections.

The subsequent step involves evaluating the performance of the proposed ILP formulation (3.6) for the projection step and identifying which type of constraints (3.2) yield superior results. A comparison was made among all proposed matching scores, excluding fused scores, as well as between greedy and exact solvers. The results including the model transfer pipeline are presented in Table 4.2. It is important to note that, given the the variables in the ILP problem are binary, certain constraint types can be omitted from testing, for example, $\sum_{p^{tgt} \in T} x_{p^{src},p^{tgt}} < n_{proj} \Leftrightarrow \sum_{p^{tgt} \in T} x_{p^{src},p^{tgt}} \leq n_{proj} - 1.$

First and foremost, the best results obtained for all languages utilizing the proposed ILP formulation surpass the highest results achieved with the heuristic word-to-word alignment-based algorithm as well as model transfer. In the case of the Italian and Spanish languages,

		tgt_lang	de		es		it	
		solver	GREEDY	GUROBI	GREEDY	GUROBI	GREEDY	GUROBI
pipeline	constr. type	n_{proj}						
align	$\leq$	1	0.920	**0.921**	0.883	0.883	0.866	0.864
		2	0.920	0.650	0.883	0.488	0.866	0.471
	$=$	1	0.920	0.918	0.883	0.883	0.866	0.864
	$\geq$	0	0.908	0.569	0.863	0.417	0.853	0.400
		1	0.908	0.569	0.863	0.417	0.853	0.400
ner	$\leq$	1	0.713	0.714	0.739	0.734	0.713	0.705
		2	0.713	0.669	0.739	0.665	0.713	0.660
	$=$	1	0.713	0.670	0.739	0.706	0.713	0.665
	$\geq$	0	0.690	0.641	0.716	0.629	0.685	0.653
		1	0.690	0.598	0.716	0.607	0.685	0.613
nmt	$\leq$	1	0.876	0.886	0.906	**0.916**	0.872	**0.879**
		2	0.876	0.602	0.906	0.635	0.872	0.601
	$=$	1	0.876	0.886	0.906	**0.916**	0.872	**0.879**
	$\geq$	0	0.233	0.180	0.315	0.220	0.271	0.203
		1	0.233	0.181	0.315	0.220	0.271	0.203
Model transfer	-	-	0.621		0.653		0.657	

Table 4.2: Overall F1 scores for the model transfer and ILP based projection pipelines on the Europarl NER dataset. Here *align* refers to the alignment-based score, *ner* denotes the NER model-based score, and *nmt* corresponds to the translation-based score.

this improved performance is attributed to the superior effectiveness of the translation-based cost, whereas alignment-based matching scores exhibit inferior results. This discrepancy can be explained by the heuristic algorithm's capability to merge aligned ranges if only one misaligned word exists between them, while the ILP formulation with alignment-based scoring can achieve similar merging only under specific circumstances. Nevertheless, for the German language, the proposed alignment-based matching score within the ILP-based projection step demonstrated superior results across all experimental evaluations including heuristics.

In the comparison between the greedy algorithm 4 and the exact solver GUROBI, it is observed that, in most cases, the greedy algorithm performs better. There are two primary reasons for this outcome. The first reason is that the exact ILP solver has to enforce constraints, leading to suboptimal solutions from an application perspective. When constraints are of the type $=$, the solver may include incorrect projections in an effort to satisfy the constraints. When constraints are ≤ 2, it tends to favor smaller, non-overlapping candidates with a higher overall cost rather than opting for a longer candidate with high individual cost. In cases where constraints are $\geq$, the solver attempts to project the source entity to as many target candidates as possible, given that all scores are non-negative. Conversely, the greedy algorithm consistently selects for projection candidates with the maximum score and does not generally adhere to the constraints (3.2) for the cases of $=, \geq, >$.

However, this does not account for why, in certain experiments involving the Italian and Spanish languages with constraints of type ≤ 1, the greedy algorithm still slightly outperforms the exact solver. This discrepancy can be attributed to the fact that all matching score calculations involve models that may themselves introduce errors, resulting in scores that are not always aligned with the application problem of projection. Nonetheless, it is noteworthy

that among the best solutions, the GUROBI optimizer outperformed the greedy algorithm in all cases.

Furthermore, it is important to note the runtimes of the various pipelines. The pipeline employing the greedy algorithm is up to two times faster than that utilizing the exact solver and is comparable to the runtime of the heuristic-based algorithm. Conversely, the runtime for the pipelines that incorporate the translation-based score, despite yielding the best metrics, is significantly longer than that of all other pipelines, as it necessitates the execution of a translation model to compute the scores. The runtimes are detailed in the Appendix A, specifically in Tables A.1 and A.2.

Another important observation is that the ILP-based projection pipeline utilizing NER model-based matching costs significantly outperforms the model transfer approach, despite the fact that the model that has been used and, consequently, the outputs of the model were identical. The improvement in F1 score can be attributed to the fact that projection pipelines needs to project source entities, thereby filtering out any predictions from the model that correspond to classes not present among the source entities. And in addition, the application of constraints (3.2) imposes limitations on the number of entities of each class, which explains the circumstances under which the metric for this type of the matching score for the ≥ 2 type of constraints appears lower to that of model transfer.

Thus, for the subsequent experiments, the constraints (3.2) will be fixed in the form of ≤ 1. Additionally, it is noted that the translation-based matching scores yield the best results and consequently should dominate, i.e. has higher weight, in all fused scores.

4.2 Intrinsic evaluation within a full pipeline

It is crucial to evaluate the proposed ILP-based projection step not only in isolated settings but also within the complete pipeline, as errors introduced by the translation and source NER labeling may accumulate and negate the advantages of the proposed formulation. Therefore, the projection step of the XLNER pipeline in the form of the ILP problem (3.6) has been evaluated using the MasakhaNER2 dataset. This dataset comprises labeled sentences in 20 African low-resource languages and classifies all entities into four categories: person (*PER*), organization (*ORG*), location (*LOC*), and date (*DATE*). However, since the source NER model employed does not support the DATE class, this particular class was disregarded during the metric computation.

English has been selected as the source language. The forward translation model used for all experiments is NLLB-200-3.3B [Tea+22]. It should be noted that this model supports only 18 of the 20 languages included in the MasakhaNER2 dataset; therefore, only these languages will be considered in our evaluation.

In comparison to the isolated evaluation of the projection step presented in the previous section, this evaluation also includes fused scores (3.12), which represent a weighted sum of the basic matching scores. All ILP-based pipelines that have been evaluated are characterized in terms of fused scores, with the corresponding weights provided in Table 4.3.

λ Pipeline	align (3.7)	ner (3.9)	ner (3.13)	nmt (3.11)
align	1	0	0	0
ner	0	1	0	0
nmt	0	0	0	1
align_ner	0.5	0.5	0	0
align_ner_spans	0.5	0	0.5	0
align_nmt	0.5	0	0	1
ner_spans_nmt	0	0	0.5	1
all_fusion	0.5	0	0.5	1

Table 4.3: A description of each tested ILP-based pipeline in a form of weights of the general fused score

The results of the evaluation of all these pipelines, along with the heuristic word-to-word alignment-based algorithm and the model transfer approach, are presented in Table 4.4. **Bold** text indicates the best overall result for a language, while underlined text signifies the best result among the pipelines featuring the ILP-based projection step. The hyperparameters for the heuristic and the type of constraints (3.2) have been selected based on the findings from the previous section, specifically corresponding to the best results from the experiments denoted in Tables 4.1 and 4.2. All experiments involving ILP-based projection pipelines were conducted using GUROBI as the exact solver to evaluate the true performance of the proposed formulation, avoiding alterations induced by the proposed approximate greedy algorithm.

tgt_lang pipeline	bam	ewe	fon	hau	ibo	kin	lug	luo	mos	nya
Model transfer	0.295	0.776	0.542	0.721	0.618	0.673	0.755	0.542	0.522	**0.802**
Heuristic	0.492	0.756	0.624	0.709	0.714	0.658	0.790	**0.742**	0.428	0.761
align	0.471	0.733	0.616	0.695	**0.736**	0.652	**0.808**	0.727	0.436	0.757
ner	0.386	0.782	0.683	**0.727**	0.635	0.687	0.781	0.610	0.518	0.759
nmt	0.245	0.760	0.612	0.713	0.623	0.665	0.778	0.641	0.401	0.729
align_ner	0.477	**0.791**	0.651	0.719	0.668	**0.689**	0.806	0.723	0.502	0.761
align_ner_spans	**0.499**	0.784	0.652	0.716	0.641	0.681	0.800	0.712	**0.523**	0.754
align_nmt	0.310	0.771	0.644	0.718	0.657	0.674	0.788	0.672	0.446	0.739
ner_spans_nmt	0.319	0.780	0.668	0.713	0.660	0.676	0.797	0.686	0.504	0.749
all_fusion	0.370	0.786	**0.684**	0.712	0.697	0.685	0.798	0.693	0.508	0.755

tgt_lang pipeline	sna	swa	tsn	twi	wol	xho	yor	zul	**avg**
Model transfer	0.365	**0.883**	0.646	0.482	0.442	0.244	0.394	0.437	0.563
Heuristic	0.678	0.792	**0.796**	**0.742**	0.583	0.526	0.489	0.641	0.662
align	0.673	0.785	0.787	0.706	0.574	0.517	0.506	0.638	0.656
ner	0.621	0.826	0.703	0.602	0.541	0.527	0.459	0.535	0.632
nmt	0.713	0.812	0.763	0.682	0.596	0.642	0.532	0.739	0.647
align_ner	0.704	0.817	0.778	0.683	0.617	0.585	0.536	0.652	0.675
align_ner_spans	0.698	0.812	0.763	0.667	0.638	0.585	0.519	0.648	0.672
align_nmt	0.722	0.811	0.772	0.692	0.631	0.646	0.565	0.741	0.667
ner_spans_nmt	0.722	0.817	0.771	0.707	0.651	**0.654**	0.552	0.740	0.676
all_fusion	**0.745**	0.817	0.776	0.711	**0.661**	0.654	**0.566**	**0.745**	**0.687**

Table 4.4: Overall F1 scores for XLNER pipelines with different projection steps on the MasakhaNER2 dataset

The results indicate that the proposed ILP-based projection step within the full XLNER pipeline performs overall better than both the best heuristic word-to-word alignment-based and model transfer approaches. This advantage is evident in both the number of languages where the proposed pipeline outperforms the others and in the average metrics across all tested languages. Specifically, for only 5 of the 18 languages, the proposed projection formulation yields worse results than the previous methods, although the gap in metrics remains small. In instances where the proposed method performs better, the differences in metrics can be significant, as illustrated by the case of isiXhosa (xho), Yorùbá (yor) and isiZulu (zul) languages.

Among the various matching scores, the *all_fusion* score, which employs a combination of all proposed matching costs, demonstrates the best results. This can be attributed to the fact that the *all_fusion* matching score integrates the advantages of each individual score, thereby providing a more accurate representation of whether a source entity should be projected onto a target candidate. In situations where the *all_fusion* score performs worse than any other ILP-based pipeline, it suggests that one of the individual compound scores make the overall fused matching score worse due to its inherent drawbacks.

Similar to the isolated evaluation, the results for the heuristic word-to-word alignment-based algorithm are slightly superior to those for the ILP-based projection utilizing alignment-based matching scores. It is attributed to the same rationale: the heuristic algorithm, in comparison to ours, always can merge two aligned word ranges that are separated by only one non-aligned word to the source entity.

For the same reasons described in the previous section, the pipeline *ner*, which utilizes the NER model-based matching score, consistently outperforms (sometimes in 1.5 times) the *Model transfer* approach when provided with the same NER model outputs. However, being projection-based, it may be susceptible to errors from the preceding steps, namely translation and source NER model labeling errors, which can adversely impact performance, as observed in the cases of the Chichewa (nya) and Kiswahili (swa) languages.

tgt_lang Method	bam	ewe	fon	hau	ibo	kin	lug	luo	mos	nya
EasyProject	0.458	0.785	0.614	0.722	0.656	0.710	0.767	0.502	0.531	0.753
CODEC	0.458	0.791	0.655	0.724	0.709	0.712	0.772	0.496	0.556	0.768
GPT-4	0.422	0.722	0.394	0.659	0.422	0.475	0.625	0.472	0.432	0.711
GoLLIE-TF	0.548	0.732	0.579	0.671	0.566	0.585	0.755	0.517	0.488	0.782
CLaP	-	-	-	0.699	0.605	-	-	-	-	0.587

tgt_lang Method	sna	swa	tsn	twi	wol	xho	yor	zul	avg
EasyProject	0.559	0.836	0.740	0.653	0.589	0.711	0.368	0.730	0.649
CODEC	0.724	0.831	0.747	0.646	0.631	0.704	0.414	0.748	0.671
GPT-4	0.395	0.792	0.563	0.442	0.526	0.498	0.547	0.369	0.526
GoLLIE-TF	0.574	0.735	0.710	0.742	0.619	0.499	0.544	0.528	0.621
CLaP	0.597	0.807	-	-	-	0.613	0.306	0.544	-

Table 4.5: Overall F1 scores for various XLNER methods evaluated under different settings compared to our experiments

Another observation involves the score (3.13), referred to as *ner_spans*, which utilizes the predictions of a NER model only to assess the likelihood that a given candidate is a span of an entity of any class. This approach leverages other scores to assign the appropriate label to this span. It demonstrates results that are comparable to those of other pipelines, especially *ner*. Notably, for the Bambara (bam) and Mossi (mos) languages, it performs the best, as it used its ability of disregarding incorrectly predicted labels for correctly predicted spans.

To assess how the performance of the proposed method compares to that of other methods described in Chapter 2, the results obtained by these methods, as reported in the original papers, are presented in Table 4.5. However, it should be noted that the evaluation setups for these experiments were different from ours, making direct comparisons not feasible. EasyProject, CODEC, and CLaP were evaluated in extrinsic settings, i.e. the reported results represent the F1 metric on the test set of the MasakhaNER2 dataset, obtained by models trained on datasets generated by these methods. GOOLIE-TF (TransFusion), GPT-4 and CLaP utilized all four classes provided in the MasakhaNER2 dataset, while our approach and others in this list were evaluated without the *DATE* class. The results for GPT4 are taken from the TransFusion paper [CSR23] and denote few-shot prompt results using GPT-4 (gpt4-02-14) with a GoLLIE [Sai+24] style prompt. Nonetheless, it is observed that the metrics fall within the same range as those from our experiments.

5 Conclusion and Further Work

In this work, we have proposed the projection step of the cross-lingual named entity recognition pipeline as the integer linear programming problem (3.6). This formulation aims to extract candidates within the target sentence and match them with the source entity provided from the preceding step of the XLNER pipeline. The likelihood of such matches is represented by the matching scores, which should be maximized under the natural constraints that the source entity cannot be projected onto overlapping candidates, and exists a limit on the number of projections for each source entity.

Three distinct options for matching costs have been proposed, each offering different relevant connections to the projection problem, including alignment-based, NER-model-based, and translation-based scores. A notable feature that differs our formulation from others is its capacity to use matching costs computed based on different principle and motivations and fuse them together to conceal the weaknesses inherent in each individual score.

For candidate extraction, a straightforward approach has been proposed, which considers all word n-grams of the target sentence constrained by the maximum length as potential candidates for matching. Furthermore, it has been proved that, for certain scores, specifically alignment-based, and constraints (3.2) of type "less" or "less or equal" this set can be reduced without compromising any optimal solution. That allows to speed up computation of solutions.

The complexity of the proposed ILP problem remains an open question. Nonetheless, it has been shown that, in the absence of constraints limiting the number of projections for each source entity, the problem can be solved by a polynomial-time algorithm. This property arises because the non-overlapping constraints, by construction, induce an interval graph. Thus, even if the problem is NP-hard, the constraints (3.2) play a critical role in its complexity.

Additionally, a fast greedy approximate algorithm has been developed for practical application, yielding solutions that satisfy non-overlapping constraints and constraints of type (3.2) when they are of form "less" or "less or equal". However, this algorithm continues to demonstrate superior performance from the application standpoint even when constraints (3.2) have other forms, as the enforcement of these constraints typically leads to less optimal solutions from an application perspective.

It is also important to highlight that the constraints restricting the number of projected candidates for each source entity, particularly with respect to constraints of type "equal", "greater", "greater or equal", may result in the optimal solution that is not aligning with the optimal labeling of the target sentence, as the exact solver has to enforce these constraints and therefore may break optimal from the application point of view projections.

We have evaluated our approach in isolation (focusing solely on the projection step) utilizing the Europarl-NER dataset, as well as within the full XLNER pipeline using the MasakhaNER2 dataset. The evaluation results indicate that the highest metrics for the application problem are attained when the constraints of type (3.2) are defined as ≤ 1. This can be attributed to the expectation of a one-to-one correspondence between source and target entities for languages sharing similar writing systems. However, due to potential errors introduced by the source NER model on the previous step of the pipeline—such as incorrect predictions of non-existent entities—and the imperfect nature of matching costs derived from various models, the solver must be permitted to not match all source entities.

Experiments demonstrate that the proposed projection method consistently outperforms heuristic word-to-word alignment-based and model transfer pipelines, particularly in scenarios involving fused scores that incorporate all types of scores, which represents a principal advantage of our method. While heuristics can occasionally outperform the ILP-based projection step utilizing alignment-based matching scores due to their aggressive filling of gaps in alignments, the proposed NER model-based score frequently exceeds the performance of model transfer that utilize the same NER model outputs, achieving improvements of up to 1.5 times. This superior performance can be attributed to the projection step's ability to filter out predicted target entities that do not correspond to any source entity.

The findings presented here advocate for further research directions. First, it is essential to determine the complexity of the proposed problem, either through reduction to or from a problem with known complexity or by providing a polynomial-time algorithm to solve this problem. Secondly, novel matching costs could be developed to extend and enhance existing ones. Particularly intriguing are matching scores that can take negative values, as this flexibility may enable the application of alternative form constraints of type (3.2), such as "greater or equal". An additional direction for research involves the refinement of candidate extraction strategies, as the current set of all n-grams up to a defined maximum length is excessively large, resulting in slower computations for both the solution and matching scores. In addition to the options discussed in the preceding sections, such as employing large language models (LLMs) or NER models to predict candidates, utilizing part-of-speech tags should also be considered, as NER entities typically correspond to noun phrases. Furthermore, it makes sense to evaluate the proposed method in settings where the source and target languages have completely different writing systems, for example for Chinese, Thai, and Japanese.

From a broader perspective, the proposed ILP problem for the projection step of the XLNER pipeline can be generalized to other Natural Language Processing tasks. For example, the word-to-word alignment problem. From its perspective, every word consists of tokens, and the task is to match words from the source sentence with words from the target sentence based on their tokens. In this context, the set of target candidates is equivalent to the set of tuples of tokens corresponding to all words in the target sentence; therefore, there is no overlap among different candidates. The remaining challenge is to derive a method for

computing matching scores. One approach can involve adopting the methodology utilized by current neural-based aligners and compute these scores based on cosine similarities between embeddings for each token.

In summary, our proposed formulation of the projection step as the ILP problem generalizes numerous previous methods and enables their integration, addressing the drawbacks of individual methods. By analyzing the solutions to the problem in conjunction with the corresponding matching scores, we gain insight into why particular entities are projected onto specific target candidates, thus enhancing interpretability, especially when compared to model transfer or the heuristic word-to-word alignment algorithm. Nevertheless, interpretability remains limited due to the fact that all proposed matching scores are computed utilizing neural network-based models.

Bibliography

[GG63] Paul C Gilmore and Ralph E Gomory. “A linear programming approach to the cutting stock problem—Part II”. In: *Operations research* 11.6 (1963), pp. 863–888.

[Dyc81] Harald Dyckhoff. “A new linear programming approach to the cutting stock problem”. In: *Operations Research* 29.6 (1981), pp. 1092–1104.

[PR91] Manfred Padberg and Giovanni Rinaldi. “A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems”. In: *SIAM Review* 33.1 (1991), pp. 60–100. DOI: 10.1137/1033004. eprint: https://doi.org/10.1137/1033004. URL: https://doi.org/10.1137/1033004.

[PB96] Madhumangal Pal and G. P. Bhattacharjee. “A sequential algorithm for finding a maximum weight *K*-independent set on interval graphs”. In: *Int. J. Comput. Math.* 60.3-4 (1996), pp. 205–214. DOI: 10.1080/00207169608804486. URL: https://doi.org/10.1080/00207169608804486.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[PS03] Sriram Pemmaraju and Steven Skiena. *Computational discrete mathematics: Combinatorics and graph theory with mathematica®*. Cambridge university press, 2003.

[TD03] Erik F. Tjong Kim Sang and Fien De Meulder. “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*. 2003, pp. 142–147. URL: https://aclanthology.org/W03-0419.

[Koe05] Philipp Koehn. “Europarl: A parallel corpus for statistical machine translation”. In: *Proceedings of machine translation summit x: papers*. 2005, pp. 79–86.

[LD10] Ailsa H Land and Alison G Doig. *An automatic method for solving discrete programming problems*. Springer, 2010.

[Bha+14] Binay K Bhattacharya et al. “Maximum Independent Set for Interval Graphs and Trees in Space Efficient Models.” In: *CCCG*. 2014.

[KB14] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: https://api.semanticscholar.org/CorpusID:6628106.

[Guo+17] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1321–1330. URL: https://proceedings.mlr.press/v70/guo17a.html.

[Vas17] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[Age+18] Rodrigo Agerri et al. "Building Named Entity Recognition Taggers via Parallel Corpora". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by Nicoletta Calzolari et al. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. URL: https://aclanthology.org/L18-1557.

[Dev+19] Jacob Devlin et al. "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)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[Bro+20] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[Con+20] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[He+20] Pengcheng He et al. "DeBERTa: Decoding-enhanced BERT with Disentangled Attention". In: *ArXiv* abs/2006.03654 (2020). URL: https://api.semanticscholar.org/CorpusID:219531210.

[Jal+20] Masoud Jalili Sabet et al. "SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 1627–1643. DOI: 10.18653/v1/2020.findings-emnlp.147. URL: https://aclanthology.org/2020.findings-emnlp.147.

[Liu+20] Yinhan Liu et al. "Multilingual Denoising Pre-training for Neural Machine Translation". In: *Transactions of the Association for Computational Linguistics* 8 (2020). Ed. by Mark Johnson, Brian Roark, and Ani Nenkova, pp. 726–742. DOI: 10.1162/tacl_a_00343. URL: https://aclanthology.org/2020.tacl-1.47.

[DN21] Zi-Yi Dou and Graham Neubig. "Word Alignment by Fine-tuning Embeddings on Parallel Corpora". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 2112–2128. DOI: 10.18653/v1/2021.eacl-main.181. URL: https://aclanthology.org/2021.eacl-main.181.

[HGC21] Pengcheng He, Jianfeng Gao, and Weizhu Chen. "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing". In: *ArXiv* abs/2111.09543 (2021). URL: https://api.semanticscholar.org/CorpusID:244346093.

[Liu+21] Linlin Liu et al. "MulDA: A Multilingual Data Augmentation Framework for Low-Resource Cross-Lingual NER". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, Aug. 2021, pp. 5834–5846. DOI: 10.18653/v1/2021.acl-long.453. URL: https://aclanthology.org/2021.acl-long.453.

[Wei+21] Gerhard Weikum et al. "Machine knowledge: Creation and curation of comprehensive knowledge bases". In: *Foundations and Trends® in Databases* 10.2-4 (2021), pp. 108–490.

[Xue+21] Linting Xue et al. "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, June 2021, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41. URL: https://aclanthology.org/2021.naacl-main.41.

[Ade+22] David Adelani et al. "MasakhaNER 2.0: Africa-centric Transfer Learning for Named Entity Recognition". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4488–4508. DOI: 10.18653/v1/2022.emnlp-main.298. URL: https://aclanthology.org/2022.emnlp-main.298.

[GAR22] Iker García-Ferrero, Rodrigo Agerri, and German Rigau. "Model and Data Transfer for Cross-Lingual Sequence Labelling in Zero-Resource Settings". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 6403–6416. DOI: 10.18653/v1/2022.findings-emnlp.478. URL: https://aclanthology.org/2022.findings-emnlp.478.

[Tea+22] NLLB Team et al. *No Language Left Behind: Scaling Human-Centered Machine Translation*. 2022. arXiv: 2207.04672 [cs.CL]. URL: https://arxiv.org/abs/2207.04672.

[VS22] Jannis Vamvas and Rico Sennrich. "NMTScore: A Multilingual Analysis of Translation-based Text Similarity Measures". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics,

Dec. 2022, pp. 198–213. DOI: 10.18653/v1/2022.findings-emnlp.15. URL: https://aclanthology.org/2022.findings-emnlp.15.

[Yan+22] Jian Yang et al. "CROP: Zero-shot Cross-lingual Named Entity Recognition with Multilingual Labeled Sequence Translation". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 486–496. DOI: 10.18653/v1/2022.findings-emnlp.34. URL: https://aclanthology.org/2022.findings-emnlp.34.

[ZGN22] Ningyu Zhang, Tao Gui, and Guoshun Nan. "Efficient and Robust Knowledge Graph Construction". In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Tutorial Abstracts*. Ed. by Miguel A. Alonso and Zhongyu Wei. Taipei: Association for Computational Linguistics, Nov. 2022, pp. 1–7. DOI: 10.18653/v1/2022.aacl-tutorials.1. URL: https://aclanthology.org/2022.aacl-tutorials.1.

[CSR23] Yang Chen, Vedaant Shah, and Alan Ritter. *Better Low-Resource Entity Recognition Through Translation and Annotation Fusion*. May 2023. DOI: 10.48550/arXiv.2305.13582.

[Che+23] Yang Chen et al. "Frustratingly Easy Label Projection for Cross-lingual Transfer". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 5775–5796. DOI: 10.18653/v1/2023.findings-acl.357. URL: https://aclanthology.org/2023.findings-acl.357.

[GAR23] Iker García-Ferrero, Rodrigo Agerri, and German Rigau. "T-Projection: High Quality Annotation Projection for Sequence Labeling Tasks". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 15203–15217. DOI: 10.18653/v1/2023.findings-emnlp.1015. URL: https://aclanthology.org/2023.findings-emnlp.1015.

[Tor+23] Sunna Torge et al. "Named Entity Recognition for Low-Resource Languages - Profiting from Language Families". In: *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*. Ed. by Jakub Piskorski et al. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1–10. DOI: 10.18653/v1/2023.bsnlp-1.1. URL: https://aclanthology.org/2023.bsnlp-1.1.

[Tou+23] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL]. URL: https://arxiv.org/abs/2307.09288.

[Zho+23] Wenxuan Zhou et al. "UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition". In: (2023). arXiv: 2308.03279 [cs.CL].

[Gur24] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2024. URL: https://www.gurobi.com.

[HTC24] Yining Huang, Keke Tang, and Meilian Chen. *Leveraging Large Language Models for Enhanced NLP Task Performance through Knowledge Distillation and Optimized Training Strategies*. 2024. arXiv: 2402.09282 [cs.CL]. URL: https://arxiv.org/abs/2402.09282.

[LMF24] Wen Lai, Mohsen Mesgar, and Alexander Fraser. "LLMs Beyond English: Scaling the Multilingual Capability of LLMs with Cross-Lingual Feedback". In: *Findings of the Association for Computational Linguistics: ACL 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 8186–8213. DOI: 10.18653/v1/2024.findings-acl.488. URL: https://aclanthology.org/2024.findings-acl.488.

[Le+24] Duong Minh Le et al. "Constrained Decoding for Cross-lingual Label Projection". In: *ArXiv* abs/2402.03131 (2024). URL: https://api.semanticscholar.org/CorpusID:267412651.

[Par+24] Tanmay Parekh et al. "Contextual Label Projection for Cross-Lingual Structured Prediction". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 5738–5757. DOI: 10.18653/v1/2024.naacl-long.321. URL: https://aclanthology.org/2024.naacl-long.321.

[Sai+24] Oscar Sainz et al. *GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction*. 2024. arXiv: 2310.03668 [cs.CL]. URL: https://arxiv.org/abs/2310.03668.

List of Figures

List of Tables

A Appendix I

This appendix presents the overall runtimes, measured in seconds, for all experiments conducted on the Europarl-NER dataset. All results were obtained on the HPC system of TU Dresden, specifically on the "Alpha Centauri" partition, where each job utilized 1 NVIDIA A100-SXM4 GPU, 64 GB of RAM, and 8 cores of AMD EPYC CPU 7352.

d	k	only_i	tgt_lang thr	de	es	it
0	1	False	0.8	10	11	13
			-	11	12	10
		True	0.8	10	13	13
			-	13	11	11
	-	False	0.8	11	9	11
			-	12	12	11
		True	0.8	10	10	11
			-	10	13	13
1	1	False	0.8	14	12	16
			-	11	10	13
		True	0.8	10	10	11
			-	10	10	10
	-	False	0.8	10	10	10
			-	10	13	11
		True	0.8	11	9	12
			-	13	10	10

Table A.1: Overall runtime is seconds for word-to-word alignments-based heuristic algorithm with different hyperparameter on the Europarl NER dataset

It should be noted that all word-to-word alignments were computed and saved prior to executing any part of the pipeline; consequently, the time required for their computation is not reflected in the tables. As a result, the comparison between the alignment-based pipeline and both the NER and NMT may not accurately represent their relative runtimes. However, it is generally anticipated that the computation of alignments will demand a duration comparable to that of executing a NER model; thus, the runtimes for the NER and alignment-based pipelines are expected to be similar.

		tgt_lang	de		es		it	
		solver	GREEDY	GUROBI	GREEDY	GUROBI	GREEDY	GUROBI
pipeline	constr. type	n_{proj}						
align	$\leq$	1	12	23	12	23	12	17
		2	12	14	12	12	12	14
	$=$	1	13	14	13	13	11	15
	$\geq$	0	13	14	12	14	12	15
		1	12	16	10	17	12	17
ner	$\leq$	1	38	60	45	68	42	58
		2	38	41	43	44	42	42
	$=$	1	38	40	42	45	42	44
	$\geq$	0	39	41	43	49	40	44
		1	38	39	42	46	42	42
nmt	$\leq$	1	742	1114	820	1576	902	1555
		2	703	790	823	894	909	969
	$=$	1	713	778	809	897	910	1050
	$\geq$	0	707	779	934	893	913	1006
		1	711	851	877	927	963	988

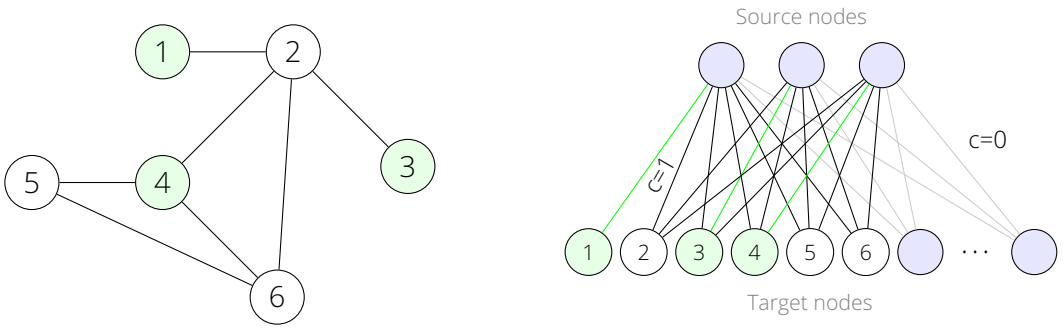
Table A.2: Overall runtime is seconds for the ILP based projection pipelines on the Europarl NER dataset

Additionally, it is worth to point out, that according to Theorem 3.4, the set of candidates T has been reduced for all ILP-based experiments that utilize only word-to-word alignments, resulting in reduced runtimes. Furthermore, the initial computation of NMT scores for the dataset (which corresponds to the *nmt* pipeline with constraints of type ≤ 1 and the GUROBI solver) caches certain precomputations. As a result, all subsequent runs involving NMT scores require less time.

B Appendix II

Consider the generalized ILP problem (3.1)–(3.4) where the relation ∩ is an arbitrary non-transitive, symmetric, and reflexive relation. In a general case, it makes the initial ILP problem harder, i.e. it can be shown that the problem (3.1)–(3.4) become an NP-hard.

The usual way to prove the complexity of some problem is to reduce other problems with known complexity to an instance of the studied one. Thus let's consider the maximum independent set problem [PS03].



(a) Maximum independent set problem (MaxIS). Vertices that form an optimal solution are colored in green

(b) Reduction to the generalized ILP problem. Dummy nodes are colored in blue

Figure B.1: The diagram of the general idea of the reduction MaxIS problem to the generalized ILP problem

The maximum independent set problem is a problem of finding a subset of vertices of some undirected graph with maximum cardinality, such that it doesn't contain any vertices connected by edges of the graph. An example of the maximum independent set problem is depicted in Figure B.1 (a). The formal definition of an ILP problem is given by B.1.

Definition B.1 (MaxIS)**.** Let $G = (V, E), V \neq \emptyset$ be an undirected graph, then a maximum independent set problem for the graph G is the following:

$$\begin{aligned} &\max \sum_{v \in V} x_v \\ &x_u + x_v \leq 1 \qquad \forall \{u, v\} \in E \\ &x_v \in \{0, 1\} \end{aligned}$$

The MaxIS problem is an NP-hard [PS03] problem since VertexCover can be reduced to an instance of this problem. But even more interesting for us is that this problem looks very similar to our projection ILP problem, except that set of constraint (3.2) is omitted and costs are equal to 1. Moreover, whether two edges of the graph are adjacent is not transitive as well as the relation of overlapping 3.2! That's why we will try to reduce the maximum independent set problem to an instance of the generalized ILP problem.

The idea of the reduction is straightforward. For every vertex $v \in V$ of the MaxIS problem we will create one distinct target candidate p_v^{tgt} that corresponds to this vertex. The set of all such target candidates will be called T_V. The ILP problem is formulated as a projection from source entities, to represent MaxIS as this ILP problem let's create dummy source nodes. The number of source nodes is determined so that it will be easy to satisfy constraints (3.2). The generalized number of dummy source nodes for any type of constraint (3.2) are given in the table B.1 (a). The matching cost between any dummy source node and target candidate from the set T_V will be equal to 1. And finally, let's link the solution of the ILP problem to the solutions of the MaxIS problem:

$$x_v = 1 \Leftrightarrow \exists p^{src} \in S \Big| x_{p^{src}, p_v^{tgt}} = 1$$

i.e. the vertex $v \in V$ belongs to the maximum independent set if and only if there is a dummy source node that are projected to the target candidate p_v^{tgt} that correspond to this vertex.

$<$	$\leq, =$	$>, \geq$
$\left\lceil \frac{\lvert V \rvert}{n_{proj}-1} \right\rceil$	$\left\lceil \frac{\lvert V \rvert}{n_{proj}} \right\rceil$	1

(a) Source nodes

$<, \leq$	$=$	$>$	$\geq$
0	$\left\lceil \frac{\lvert V \rvert}{n_{proj}} \right\rceil n_{proj}$	n_{proj}	$n_{proj}+1$

(b) Target nodes

Table B.1: Number of dummy nodes for every type of constraints (3.2)

We will consider that target nodes $t_u, t_v \in T_V$ that correspond to graph's nodes $u, v \in V$ overlap if and only if there is an edge in the graph G between these nodes or $u = v$.

$$t_u \cap t_v \neq \emptyset \Leftrightarrow (\{u, v\} \in E) \vee (u = v) \tag{B.1}$$

Such a definition of overlapping allows us to match all properties, i.e. reflexivity, symmetricity, and nontransitivity, of overlapping relation on word ranges in the initial definition 3.2. The fact that we consider overlapping nodes that correspond to the same vertex simply means that it is impossible to add the same vertex in the independent set two times which is a property of any set.

Hence the resulting ILP problem induced by the MaxIS problem is the following:

$$\begin{aligned}
&\max_x \sum_{(p^{src}, p^{tgt}) \in S \times T} c_{p^{src}, p^{tgt}} x_{p^{src}, p^{tgt}} \\
&\text{subject to} \\
&x_{p_1^{src}, p_1^{tgt}} + x_{p_2^{src}, p_2^{tgt}} \leq 1 && \forall (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S, T) \\
&x_{p^{src}, p^{tgt}} \in \{0, 1\} && \forall (p^{src}, p^{tgt}) \in S \times T
\end{aligned} \tag{B.2}$$

where set $\hat{\Pi}$ defined based on the introduced above relation $\cap$.

The problem (B.2) is exactly the same as the ILP problem (3.1)–(3.4) except the fact that constraints (3.2) are omitted. The reason is that we can not be sure that they can be satisfied for the type of constraints with $=, >, \geq$ inequalities since it is possible to just run out of nodes. To overcome this problem, let's create dummy target nodes. The generalized minimum number of such nodes is given in the table B.1 (b). Let's denote the set of all such nodes as T_{dummy} and then the set of all target candidates will be $T = T_V \cup T_{dummy}$. We will consider that dummy target nodes don't overlap with any node except itself:

$$\forall p_1^{tgt} \in T_{dummy},\ p^{tgt} \in T \quad p_1^{tgt} \cap p_2^{tgt} \neq \emptyset \Leftrightarrow p_1^{tgt} = p_2^{tgt}$$

Target dummy nodes are created only to satisfy constraints (3.2), so all matching scores between any dummy source node and any dummy target node set to be equal 0.

Since we have a one-to-one correspondence between vertices from the MaxIS problem and the target candidates from the set T, consider the following set that fully determines the solution of the MaxIS problem:

$$T_x^* = \{t \in T | \exists s \in S, x_{s,t} = 1\}$$

Let's x_1, x_2 be a two feasible solutions of problem (3.14), then we will say that they are in relation $\sim$ if their corresponding solutions of the MaxIS problem are equal:

$$x_1 \sim x_2 \Leftrightarrow T_{x_1}^* = T_{x_2}^*$$

Since this relation is defined by equality of sets, it is reflexive, symmetric, and transitive and therefore equivalence relation.

Then quotient set $X/_\sim$ will consist of all equivalence classes of feasible solutions that differ only in matching with dummy nodes. Every such equivalence class corresponds to one feasible solution of the maximum weight-independent set problem.

Let's notice that the objective function value is equal for all elements within any equivalence class.

Lemma B.1. *Let X be a set of feasible solutions of the problem* (B.2)*. Then for any equivalence class in the quotient set* $X/_\sim$ *the objective value is the same for every solution within this equivalence class.*

Proof. Consider an equivalence class $[x]$ of some feasible solution x. Let's notice that non-zero scores have only elements from the set T_x^*, therefore the objective determines by this set of projected target candidates:

$$\sum_{(p^{src},p^{tgt}) \in S \times T} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} = \sum_{(p^{src},p^{tgt}) \in S \times T_V} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} = \sum_{(p^{src},p^{tgt}) \in S \times T_x^*} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}}$$

Because of non-overlapping constraints and the fact that we consider that the target candidate overlaps with itself only one source entity can be projected onto every target candidate from the set T_x^*:

$$\forall p^{tgt} \in T_x^*,\ \exists! p^{src} \in S \Big| x_{p^{src},p^{tgt}} = 1$$

Then the objective value equals the cardinality of the set T_x^* that are equal for any solution without the same equivalence class by the definition of the relation .

$$\sum_{(p^{src},p^{tgt}) \in S \times T} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} = \sum_{(p^{src},p^{tgt}) \in S \times T_x^*} c_{p^{src},p^{tgt}} x_{p^{src},p^{tgt}} = |T_x^*|$$

□

Corollary B.1.1. *The objective function value of the ILP problem* (B.2) *and the maximum independent set problems are equal.*

Proof. By construction, every target candidate from the set T_x^* corresponds to only one vertex from the MaxIS problem. Therefore the objective value is equal to the cardinality of the set T_x^*. In the proof of the lemma, we showed that the objective value of the ILP problem (B.2) equals to $|T_x^*|$ as well. □

Dummy source nodes exist only to be able to formulate the MaxIS problem as the desired ILP projection problem. Hence we can swap a dummy source node for a target candidate it was projected from.

Lemma B.2. *Let x be a feasible solution of the problem* (B.2) *such that some dummy source node $p^{src} \in S$ is projected onto $p^{tgt} \in T$, i.e. $x_{p^{src},p^{tgt}} = 1$. Then for any other source node $\hat{p}^{src} \in S, p^{src} \neq \hat{p}^{src}$ a solution x^*, such that*

$$\begin{aligned} &x^*_{\hat{p}^{src},p^{tgt}} = 1 \qquad x^*_{p^{src},p^{tgt}} = 0 \\ &\forall (s,t) \in S \times T \setminus \{(p^{src},p^{tgt}),(\hat{p}^{src},p^{tgt})\} \quad x^*_{s,t} = x_{s,t} \end{aligned}$$

is also feasible and in the same equivalence class as x.

Proof. Suppose that the solution x^* is not feasible, it means that:

$$\exists s \in S, t \in T \Big| (s,\hat{p}^{src},t,p^{tgt}) \in \hat{\Pi}(S,T) \quad x^*_{s,t} = 1$$

Let's notice that $s \neq p^{src}$, since $x^*_{p^{src},p^{tgt}} = 0$. But then by construction:

$$x_{p^{src},p^{tgt}} + x_{s,t} = 1 + x^*_{s,t} = 1 + 1 > 2$$

It contradicts with the fact that x is a feasible solution, therefore x^* should be feasible.

And since by construction sets T_x^* and $T_{x^*}^*$ are equal solutions are in the same equivalence class and by the corollary B.1.1 the objective values for these solutions are also equal. □

Corollary B.2.1. *For any feasible solution of the problem* (B.2) *there is a feasible solution* x^* *such that*

$$\sum_{p^{tgt} \in T} x^*_{p^{src}, p^{tgt}} \leq \frac{|V|}{|S|} \qquad \forall p^{src} \in S$$

and $T^*_{x^*} = T^*_x$.

Proof. Assume the solution $\hat{x}$ such that

$$\forall p^{src} \in S \;\; \forall p^{tgt} \in T_V \qquad \hat{x}_{p^{src}, p^{tgt}} = x_{p^{src}, p^{tgt}}$$
$$\forall p^{src} \in S \;\; \forall p^{tgt} \in T_{dummy} \qquad \hat{x}_{p^{src}, p^{tgt}} = 0$$

By construction, this solution has the same objective value since all scores for dummy target nodes are equal to 0, the same set $T^*_{\hat{x}} = T^*_x$ and also it doesn't violate the non-overlapping constraints:

$$\forall (p^{src}_1, p^{src}_2, p^{tgt}_1, p^{tgt}_2) \in \hat{\Pi}(S,T) \Big| p^{tgt}_2 \in T_V \quad \hat{x}_{p^{src}_1, p^{tgt}_1} + \hat{x}_{p^{src}_2, p^{tgt}_2} = x_{p^{src}_1, p^{tgt}_1} + x_{p^{src}_2, p^{tgt}_2} \leq 1$$
$$\forall (p^{src}_1, p^{src}_2, p^{tgt}_1, p^{tgt}_2) \in \hat{\Pi}(S,T) \Big| p^{tgt}_2 \in T_{dummy} \quad \hat{x}_{p^{src}_1, p^{tgt}_1} + 0 \leq x_{p^{src}_1, p^{tgt}_1} + x_{p^{src}_2, p^{tgt}_2} \leq 1$$

Because of non-overlapping constraints (3.3) at most one source node can be projected on every target node, therefore:

$$\sum_{p^{src} \in S} \sum_{p^{tgt} \in T} \hat{x}_{p^{src}, p^{tgt}} \leq |T_V| \leq |V|$$

Suppose there exists such a $p^{src} \in S$ that

$$\sum_{p^{tgt} \in T} \hat{x}_{p^{src}, p^{tgt}} > \frac{|V|}{|S|}$$

Then there is another source entity $\hat{p}^{src}$ for which we have

$$\sum_{p^{tgt} \in T} \hat{x}_{\hat{p}^{src}, p^{tgt}} < \frac{|V|}{|S|}$$

By the lemma B.2 there exists such an optimal solution x^* where

$$\sum_{p^{tgt} \in T} x^*_{p^{src}, p^{tgt}} = \sum_{p^{tgt} \in T} \hat{x}_{p^{src}, p^{tgt}} - 1$$
$$\sum_{p^{tgt} \in T} x^*_{\hat{p}^{src}, p^{tgt}} = 1 + \sum_{p^{tgt} \in T} \hat{x}_{\hat{p}^{src}, p^{tgt}}$$

and $T^*_x = T^*_{\hat{x}} = T^*_{x^*}$. If this solution doesn't satisfy the desired property repeat the procedure. □

Lemma B.3. *Suppose* x_v *is a feasible solution of the maximum weight independent set problem, then there is a corresponding feasible solution of the problem* (B.2).

Proof. Suppose some non-feasible solution x that corresponds to the feasible solution of the MaxWIS problem. Then we have:

$$\exists(p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S,T) \Big| x_{p_1^{src},p_1^{tgt}} + x_{p_2^{src},p_2^{tgt}} > 1$$

By construction of the set $\hat{\Pi}(S,T)$ and the overlapping relation (dummy target nodes are overlapping only with itself) if $p_1^{tgt} \neq p_2^{tgt}$ then:

$$\Big\{(p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \Big| (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S,T)\Big\} = \\ \Big\{(p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \Big| (p_1^{src}, p_2^{src}, p_1^{tgt}, p_2^{tgt}) \in \hat{\Pi}(S,T_V)\Big\}$$

But since the solution x_v is feasible constraints with $p_1^{tgt} \neq p_2^{tgt}$ can not be violated:

$$\forall\{u,w\} \in E \quad x_u + x_w \leq 1 \overset{\text{(B.1)}}{\implies} \\ \forall p_1^{tgt} \in T_x^* \nexists p_2^{tgt} \in T_x^* \Big| p_1^{tgt} \cap p_2^{tgt} \neq \emptyset \implies \\ \forall p_1^{src}, p_2^{src} \in S \quad x_{p_1^{src},p_1^{tgt}} + x_{p_2^{src},p_2^{tgt}} \leq 1$$

Therefore $p_1^{tgt} = p_2^{tgt}$ and constraints are violated because there exists at least two source nodes $p_1^{src}, p_2^{src} \in S$ that are projected onto the same target nodes $p^{tgt} \in T$:

$$x_{p_1^{src},p^{tgt}} = 1 \qquad x_{p_2^{src},p^{tgt}} = 1$$

But then consider a solution x^* such that we remove one of the projections that make this constraint violated:

$$\forall(s,t) \in S \times T \setminus \{(p_2^{src}, p^{tgt})\} \quad x^*_{s,t} = x_{s,t} \\ x^*_{p_2^{src},p^{tgt}} = 0$$

Note that $T^*_{x^*} = T^*_x$ because there is a projection to the target candidate p^{tgt} since $x^*_{p_1^{src},p^{tgt}} = 1$. If the solution x^* is infeasible - repeat the procedure, otherwise this solution is feasible and corresponds to the same solution x_v of the MaxWIS problem. □

And finally, we can prove the complexity of the generalized ILP problem by reducing the MaxIS problem to an instance of (3.1)–(3.4).

Theorem B.4 ($MaxIS \leq_P ILP_{proj}$). *The generalized ILP problem* (3.1)–(3.4) *is NP-hard*

Proof. Assume the reduction of the MaxIS problem to an instance of the projection ILP problem described above. Since the number of nodes in the ILP problem is linear on the number of vertices in the MaxIS problem and has a solution to the ILP problem it scales at most quadratically on a number of vertices time to determine the solution of the MaxIS problem the reduction takes a polynomial time. The only thing that is required to check is whether the projected optimal solution to the ILP problem will be always the optimal solution of the MaxIS and vice versa.

By the corollary B.1.1 the objective values of two problems are equal and therefore feasible solution with the highest objective value will have the highest objective value in the counterpart problem. Then by lemmas B.3 and 3.8 the non-overlapping constraint won't make any optimal solution infeasible for their counterpart.

But besides non-overlapping constraint the problem (3.1)–(3.4) also has constraints (3.2). Let's check whether they won't make any optimal solution of the MaxIS problem infeasible in their ILP formulation and if there are no new optimal solutions.

Consider a set X of all feasible solutions of the problem (B.2) and set Y of all feasible solutions of the problem (3.1)–(3.4). Since the latter problem is constrained version of the problem (B.2) $Y \subset X$, but then:

$$Y \subset X \implies \forall y \in Y \quad \exists x \in X \Big| \quad [y] \subset [x],$$

i.e. there won't be any new equivalence classes and therefore no new feasible solutions of the MaxIS problem. But it turns out that for any equivalence class there is a feasible solution of the full generalized problem that belongs to this class:

$$\forall x \in X \quad \exists y \in Y \Big| \quad y \in [x]$$

Let's prove it. For this, we need to analyze constraints (3.2) for every type of inequality.

The case of $<$. The number of dummy source nodes is given in the table B.1 (a) and equal to $|S| = \left\lceil \frac{|V|}{n_{proj}-1} \right\rceil$ By the corollary B.2.1 there exists such a solution x^* from the same equivalence class that satisfies constraints:

$$\forall p^{src} \in S \quad \sum_{p^{tgt} \in T} x^*_{p^{src},p^{tgt}} \leq \frac{|V|}{|S|} = \frac{|V|}{\left\lceil \frac{|V|}{n_{proj}-1} \right\rceil} \leq \frac{|V|}{\frac{|V|}{n_{proj}-1}} = n_{proj} - 1 < n_{proj}.$$

The case of $\leq$. By the same corollary B.2.1 as in the previous case we obtain that there is a solution x^* that satisfies constraints:

$$\forall p^{src} \in S \quad \sum_{p^{tgt} \in T} x^*_{p^{src},p^{tgt}} \leq \frac{|V|}{|S|} = \frac{|V|}{\left\lceil \frac{|V|}{n_{proj}} \right\rceil} \leq \frac{|V|}{\frac{|V|}{n_{proj}}} = n_{proj}.$$

The case of $=$. By the corollary B.2.1 for any solution x from the equivalence class $[x]$ there is a feasible solution $x^* \in [x]$ such that:

$$\forall p^{src} \in S \quad \sum_{p^{tgt} \in T} x^*_{p^{src},p^{tgt}} \leq \frac{|V|}{|S|} = \frac{|V|}{\left\lceil \frac{|V|}{n_{proj}} \right\rceil} \leq \frac{|V|}{\frac{|V|}{n_{proj}}} = n_{proj}. \tag{B.3}$$

If constraints (3.2) are violated then there is such a source node $\hat{p}^{src}$ for which the sum is a strong inequality:

$$\exists \hat{p}^{src} \in S \quad \sum_{p^{tgt} \in T} x^*_{p^{src},p^{tgt}} \leq \frac{|V|}{|S|} < n_{proj}. \tag{B.4}$$

But then we can find such a dummy target node $t_d \in T_{dummy}$ that:

$$\exists t_d \in T_{dummy} \Big| \forall s \in S \quad x^*_{s,t} = 0$$

It can be proved by contradiction to the property of the given by the corollary B.2.1 and that there exists such a $\hat{p}^{src}$:

$$\begin{aligned}
&\nexists t_d \in T_{dummy} \Big| \forall s \in S \quad x^*_{s,t} = 0 \implies \\
&\forall t \in T_{dummy},\ \exists s \in S \quad x^*_{s,t} = 1 \implies \\
&\left\lceil \frac{|V|}{n_{proj}} \right\rceil n_{proj} = |S| n_{proj} \overset{\text{(B.3)}}{\geq} \sum_{s \in S} \sum_{t \in T} x^*_{s,t} \geq |T_{dummy}| = \left\lceil \frac{|V|}{n_{proj}} \right\rceil n_{proj} \implies \\
&\sum_{s \in S} \sum_{t \in T} x^*_{s,t} = \left\lceil \frac{|V|}{n_{proj}} \right\rceil n_{proj} \geq |V| \implies \\
&\sum_{s \in S \setminus \{\hat{p}^{src}\}} \sum_{t \in T} x^*_{s,t} = \left\lceil \frac{|V|}{n_{proj}} \right\rceil n_{proj} = |S| n_{proj} \overset{\text{(B.4)}}{\implies} \\
&\exists s \in S \quad \sum_{p^{tgt} \in T} x^*_{p^{src}, p^{tgt}} > \frac{|V|}{|S|} > n_{proj}
\end{aligned}$$

In short words always there are a "free" dummy target nodes that can be used to ensure equality constraints. It was expected just because during construction we chose the number of target dummy nodes to make it possible. To make the solution closer to being feasible we just need to project source entity $\hat{p}^{src}$ onto this target candidate t_d. So, the modified solution $\hat{x}$ is the following:

$$\begin{aligned}
&\forall (s,t) \in S \times T \setminus \{(\hat{p}^{src}, t_d)\}\ \hat{x}_{s,t} = x^*_{s,t} \\
&\hat{x}_{\hat{p}^{src}, t_d} = 1
\end{aligned}$$

By the definition of the set $T^*_{\hat{x}}$ since we add only a projection to a dummy node it won't change the equivalence class of the solution and objective value, but make the number of projection from the source entity $\hat{p}^{src}$ higher:

$$\sum_{p^{tgt} \in T} \hat{x}_{\hat{p}^{src}, p^{tgt}} = \sum_{p^{tgt} \in T} x^*_{\hat{p}^{src}, p^{tgt}} + 1 \leq n_{proj}$$

Since dummy target nodes overlap only with itself we have the constraints that potentially can be violated are defined by this set:

$$\begin{aligned}
\left\{ (p^{src}_1, p^{src}_2, t_d, p^{tgt}_2) \Big| (p^{src}_1, p^{src}_2, p^{tgt}_1, p^{tgt}_2) \in \hat{\Pi}(S,T) \right\} = \\
\left\{ (p^{src}_1, p^{src}_2, t_d, t_d) \Big| p^{src}_1, p^{src}_2 \in S, p^{src}_1 \neq p^{src}_2 \right\}
\end{aligned}$$

But because of the choice of t_d the constraints (3.3) are not violated:

$$\begin{aligned}
\forall p^{src}_1, p^{src}_2 \in S \Big| p^{src}_1 \neq p^{src}_2, p^{src}_1 \neq \hat{p}^{src}, p^{src}_2 \neq \hat{p}^{src} \quad \hat{x}_{src p_1, t_d} + \hat{x}_{p^{src}_2, t_d} = x_{p^{src}_1, t_d} + x_{p^{src}_2, t_d} = 0 \\
\forall p^{src}_2 \in S \Big| \hat{p}^{src} \neq p^{src}_2 \quad \hat{x}_{\hat{p}^{src}, t_d} + \hat{x}_{p^{src}_2, t_d} = 1 + 0 = 1 \leq 1
\end{aligned}$$

If still there are some source entities for which constrains (3.2) are violated - repeat the process. Repetition of the process is possible since the property (B.3) still holds and we proved that there always exists a "free" dummy target node we can project this source node onto.

The case of $>, \geq$. We can perform a similar derivation as for the case of $=$ type of constraints, but let's note that $\forall p^{src} \in S$:

$$\sum_{p^{tgt} \in T} x_{p^{src}, p^{tgt}} = n_{proj} \Leftrightarrow n_{proj} \geq \sum_{p^{tgt} \in T} x_{p^{src}, p^{tgt}} \geq n_{proj}$$

$$\bigvee_{k=n_{proj}+1}^{|T|} \sum_{p^{tgt} \in T} x_{p^{src}, p^{tgt}} = k \Leftrightarrow \sum_{p^{tgt} \in T} x^*_{p^{src}, p^{tgt}} > n_{proj}$$

Therefore all solutions that are feasible for the case of equality constraints are also should be feasible for the case of $\geq, >$ and vice versa, so we won't lose any equivalence class.

So, we have shown that for every feasible solution of the problem (B.2) there is a feasible solution of the full generalized ILP problem (3.1)–(3.4) that belongs to the same equivalence class. And every feasible solution of the full generalized ILP problem is a feasible solution of the problem (B.2). Therefore the MaxIS problem can be solved as an instance of the problem (3.1)–(3.4) and consequently, the generalized ILP problem is at least as hard as the maximum independent set problem which is NP-hard. □

Acknowledgement

This work is inspired by the current research on cross-lingual entity recognition conducted at ScaDS.AI (Center for Scalable Data Analytics and Artificial Intelligence) in Dresden/Leipzig, where the author is employed. However, the idea for this work was independently proposed and developed by the author.

The author gratefully acknowledges the computing time made available on the high-performance computer at the NHR Center of TU Dresden. This center is jointly supported by the Federal Ministry of Education and Research and the state governments participating in the NHR (www.nhr-verein.de/unsere-partner)