


Jožef Stefan International Postgraduate School
La Rochelle University
Euclide Doctoral School
Laboratory of Informatics, Image and Interaction (L3i)

Doctoral Dissertation presented by:

Hanh Thi-Hong TRAN

submitted in : **August 2024**

to be defended on : **26 September 2024** in Ljubljana

to obtain the titles: **Doctor of Philosophy from La Rochelle University**

(Discipline: Information Technology and Applications) &

Doctor of Philosophy from Jožef Stefan International Postgraduate School

(Discipline: Information and Communication Technology)

NEURAL APPROACHES TO AUTOMATIC TERMINOLOGY EXTRACTION

MARKO ROBNIK ŠIKONJA	PROFESSOR	Ljubljana University, Slovenia	Reviewer
FLORIAN BOUDIN	ASSOCIATE PROFESSOR	Nantes University, France	Reviewer
ELS LEFEVER	ASSOCIATE PROFESSOR	Ghent University, Belgium	Reviewer
ADAM JATOWT	PROFESSOR	Innsbruck University, Austria	Examiner
SENJA POLLAK	ASSISTANT PROFESSOR	Jožef Stefan Institute, Slovenia	Co-Director
ANTOINE DOUCET	PROFESSOR	La Rochelle University, France	Co-Director






Jožef Stefan International Postgraduate School
La Rochelle University
Euclide Doctoral School
Laboratory of Informatics, Image and Interaction (L3i)

Hanh Thi-Hong TRAN

Doctoral dissertation

NEURAL APPROACHES TO AUTOMATIC TERMINOLOGY EXTRACTION

Thèse de doctorat

APPROCHES NEURONALES DE L'EXTRACTION AUTOMATIQUE DE TERMINOLOGIE

Doktorska disertacija

NEVRONSKI PRISTOPI K SAMODEJNEMU LUŠČENJU TERMINOLOGIJE

Supervisors: Asst. Prof. Dr. Senja Pollak & Prof. Dr. Antoine Doucet

La Rochelle, France, August 2024

Ljubljana, Slovenia, August 2024

NEURAL APPROACHES TO AUTOMATIC TERMINOLOGY EXTRACTION

Hanh Thi-Hong Tran

Doctoral Dissertation
Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
La Rochelle University, La Rochelle, France

Supervisor: Asst. Prof. Dr. Senja Pollak, Jožef Stefan Institute, Slovenia
Supervisor: Prof. Dr. Antoine Doucet, La Rochelle University, France

Evaluation Board:
Prof. Marko Robnik Šikonja, Chair, University of Ljubljana, Slovenia
Assoc. Prof. Florian Boudin, Member, Nantes University, France
Assoc. Prof. Els Lefever, Member, Ghent University, Belgium

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Hanh Thi-Hong Tran

NEURAL APPROACHES TO AUTOMATIC TERMINOLOGY EXTRACTION

Doctoral Dissertation

NEVRONSKI PRISTOPI K SAMODEJNEMU LUŠČENJU TERMINOLOGIJE

Doktorska disertacija

Supervisor: Asst. Prof. Dr. Senja Pollak

Supervisor: Prof. Dr. Antoine Doucet

Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
La Rochelle University, La Rochelle, France, August 2024

Acknowledgments

I would like to take this moment to convey my deepest gratitude to the numerous individuals who have provided unwavering support and shared their insights throughout the journey of my doctoral thesis.

Foremost, I extend my deepest gratitude to my esteemed mentors, Prof. Antoine Doucet and Asst. Prof. Senja Pollak. Their invaluable guidance, astute feedback, and constructive comments have been the cornerstone of my research. Their expertise and unwavering dedication have not only steered the course of my work but also served as a perpetual wellspring of inspiration.

A special note of appreciation goes out to the participants of the Slovene CANDAS and the French TERMITRAD projects. To Matej Martinc, Andraz Pelicon, Boshko Koloski, and Jaya Caporusso from the Jožef Stefan Institute (JSI) team, and Carlos Gonzalez Gallardo, Wenjun Sun, Julien Delaunay, and Nicolas Sidère from the La Rochelle team, I extend my gratitude for our collaborative brainstorming sessions, unwavering support, and teamwork. Many milestones in this journey would not have been reached without our exceptional teamwork.

I would like to acknowledge the indispensable contributions of colleagues: Mili Bauer, Živa Antauer, Syrielle Montariol, Sebastian Mežnar, Nina Omejc, Boštjan Gec, Sintija Stevanoska from the E8 Department, JSI; Nguyen Tien Nam, Pham Tri Cong and his family, Eliott Thomas, Dipendra Sharma Kafle, Baglan Aitu, Laetitia Barreau, and Muzzamil Luqman at L3i Lab; Isabelle, and Jenifer from École Doctorale. Their support and shared experiences have infused this academic pursuit with enjoyment and significance.

Lastly, I am profoundly indebted to my cherished family. To my father, Tran Van Thanh, my mother, Nguyen Thi Hong, my father-in-law, Ngoc Hung, and my mother-in-law, Nguyen Thi Huong, as well as my beloved husband, Ngoc Viet Tien, and our endearing feline companion, Do Do, I offer my deepest gratitude for their enduring love, encouragement, and support throughout my academic journey. Their unshakable belief in my abilities and the sacrifices they made to facilitate my education is immeasurable.

While it is impossible to acknowledge everyone who has contributed in some way to this thesis, I want to express my sincere thanks to the friends I had the privilege of meeting in Ljubljana and La Rochelle, including Taisiia Medviedieva, Eva Šalamun Trojar, Miha Nguyen, and Vietnamese family at Dobro Jutro Vietnam (Ljubljana) restaurant. Their encouragement and assistance have been invaluable to me.

Abstract

Automatic terminology extraction, also known as automatic term extraction (ATE), is a natural language processing (NLP) task that identifies specialized terminology from domain-specific corpora. ATE is often used for terminographic tasks (e.g., the creation of specialized dictionaries) and contributes to several complex downstream tasks (e.g., machine translation and information retrieval). Over the last forty years, considerable progress has been made in automatic terminology extraction, however, several major challenges persist.

At the beginning of our studies, most ATE systems relied on either shallow machine learning (ML) or deep neural networks (e.g., transformers). While the earlier techniques suffered from time-consuming feature engineering steps and difficulties in generalizing to new unseen domains, the later approaches solved these problems by introducing the task as a binary sequence classification problem with transformers variants. However, generating all possible n-grams from each sentence across all documents for training purposes still poses a computational challenge. Moreover, current systems only focus on developing a system to extract the non-nested terms in fully supervised environments, leaving a gap in capturing nested terms and handling scenarios where there is not enough data. Therefore, in our thesis, we address these challenges in ATE.

We focus on the following aspects. First, in scenarios where sufficient annotated data is available for fully supervised settings, we investigate the improvement of neural approaches by introducing the task as a token classification problem (so-called sequence labeling), using transformers as a base model with additional representation (e.g., label semantics) and modified layers (e.g., mixture of experts or MoE, RNN). Furthermore, we build on the current systems by introducing NOBI, a novel annotation system to better capture the nested terms. Second, in scenarios where the well-annotated data from the same language is limited but the data from other languages is suitable for fully supervised settings, we propose cross-lingual and multilingual learning to exploit the potential of transfer learning between languages, especially for languages with fewer resources. Third, in scenarios where both well-annotated data and computational resources are limited, we propose a novel pipeline called *LlamATE* that uses large language models (LLMs) as a predictor to query the candidate terms without additional fine-tuning steps, using only demonstrations with few shot and self-verification steps.

Our study comes to the following conclusions. First, token classification approaches (e.g., using XLMR) are valid and promising methods for fully supervised learning in terminology extraction, as they outperform non-sequential and binary sequence classifiers and reduce the computational challenges mentioned in the benchmarks. Integrating a MoE layer on top of the deep neural model (e.g., (m)DeBERTA) improves performance compared to the baseline with a dense token classification head. Using NOBI annotation regimes for the token classifier trained on the datasets with a significant amount of nested terms shows a visible improvement on single-word terms. Second, our results on token classification on limited annotated data for a given language demonstrate the

promising effect of multilingual and cross-lingual cross-domain learning, which is particularly important when transferring from the rich- to lesser-resourced languages. Finally, our pipeline, *LlamATE*, suggests the potential of LLMs with few-shot demonstrations and self-verification in learning from a few examples in the same domain even without explicitly naming the domain, as well as the potential in transferring knowledge from well-covered languages (i.e., English) to less-represented languages in pre-trained LLMs (i.e., French, Dutch). Even though the LLM-based approaches with few-shot demonstrations are not a substitute for fully supervised models, they can provide solutions with relatively good performance without the need for manually annotated data.
Keywords: Automatic terminology extraction, Transformers, Token Classification, Seq2seq, Large Language Model, Prompt Engineering, In-context Learning, Llama2, ChatGPT.

Résumé

L'extraction automatique de terminologie (EAT) est une tâche de traitement automatique du langage naturel (TALN) qui identifie la terminologie spécialisée à partir de corpus spécifiques à un domaine. En réduisant le temps et les efforts nécessaires à l'extraction manuelle des termes, l'EAT est non seulement largement utilisée pour des tâches terminologiques (par exemple, la création de dictionnaires spécialisés) mais contribue également à plusieurs tâches complexes en aval (par exemple, la traduction automatique et la recherche d'informations). Au cours des quarante dernières années, des progrès considérables ont été réalisés pour fournir automatiquement une liste classée des termes candidats à partir de corpus spécialisés; cependant, les tâches d'EAT restent un problème notoirement difficile.

Au début de notre étude, la plupart des systèmes d'EAT reposaient soit sur des techniques d'apprentissage automatique simples, soit sur des réseaux de neurones profonds. Alors que les premières techniques souffrent d'étapes d'ingénierie des caractéristiques longues et de la difficulté de généralisation à de nouveaux domaines invisibles, les approches ultérieures ont surpassé ces goulets d'étranglement en introduisant la tâche comme un problème de classification de séquence binaire. Cependant, générer tous les n-grammes possibles à partir de chaque phrase de tous les documents à des fins d'entraînement pose toujours des défis informatiques et de stockage. De plus, les systèmes actuels se concentrent uniquement sur la construction d'un système pour extraire les termes non imbriqués dans des paramètres entièrement supervisés, laissant un vide dans la capture des termes imbriqués et dans la gestion des scénarios où il y a un manque de données bien annotées dans les mêmes langues. Par conséquent, nous pensons qu'il reste une marge d'amélioration dans le domaine de l'extraction de termes supervisée.

Pour relever les défis ci-dessus, notre thèse s'est concentrée sur les aspects suivants. En ce qui concerne les scénarios où les données bien annotées sont adéquates pour des paramètres entièrement supervisés, nous avons étudié l'amélioration des approches neuronales en introduisant la tâche comme un problème de classification de jetons (appelé étiquetage de séquence) en utilisant des Transformers comme modèle de base avec une représentation supplémentaire (par exemple, la sémantique des étiquettes) et des couches modifiées (par exemple, mélange d'experts, RNN). De plus, nous avons développé les systèmes actuels en introduisant NOBI, un nouveau régime d'annotation pour mieux capturer les termes imbriqués. En ce qui concerne les scénarios où les données bien annotées des mêmes langues sont limitées, nous avons proposé un apprentissage cross-lingue et multilingue pour mettre en évidence le potentiel du transfert d'apprentissage des langues à ressources riches ou combinées vers des langues moins dotées dans les systèmes neuronaux. En ce qui concerne les scénarios où les données bien annotées et les ressources de calcul sont limitées, nous avons proposé un nouveau pipeline utilisant des modèles de langage volumineux (LLM) appelés LlamATE comme prédicteur pour interroger les termes candidats sans aucune étape de réglage fin supplémentaire, en utilisant simplement une démonstration par few-shot avec des étapes d'auto-vérification.

Notre étude s'est terminée par les conclusions suivantes. Premièrement, les approches de classification de jetons (par exemple, XLMR) se sont révélées des méthodes valides et prometteuses pour l'apprentissage entièrement supervisé en EAT. Ces approches surpassent les performances des classifieurs non séquentiels et à séquence binaire, et réduisent les défis de calcul et de stockage mentionnés dans les tests de performance. L'ajout d'une couche MoE au-dessus du modèle neuronal profond (par exemple, (m)DeBERTA) a constamment amélioré les performances par rapport à la configuration de base avec une tête de classification de jetons dense. L'utilisation des régimes d'annotation NOBI a démontré une amélioration visible pour les termes en un seul mot et multi-mots du classifieur de jetons entraîné sur le jeu de données dans lequel le nombre de termes imbriqués est suffisamment important. Deuxièmement, avec des données annotées limitées provenant des mêmes langues, nos résultats sur le classificateur de jetons ont mis en évidence l'impact prometteur de l'apprentissage interdomaine multilingue et interlinguistique lors du transfert des connaissances des langues riches vers des langues moins connues. Enfin, notre pipeline nommé LlamATE suggère le potentiel des LLM avec démonstration par few-shot et auto-vérification pour apprendre à partir de quelques exemples dans le même domaine, même sans que le domaine ne soit explicitement indiqué. Il suggère également le potentiel de transfert de connaissances des langues bien couvertes (anglais) vers des langues moins représentées (français, néerlandais) dans les LLM. Bien qu'ils ne remplacent peut-être pas entièrement les modèles entièrement supervisés, ils peuvent améliorer l'efficacité et la précision en rationalisant le processus de pré-annotation et en accélérant les efforts d'annotation manuelle.

Mots-clés: Extraction automatique de terminologie, Transformers, Classification de jetons, Seq2seq, Modèle de langage volumineux, Ingénierie des invites, Apprentissage en contexte, Llama2, ChatGPT.

Povzetek

Samodejno luščenje terminologije (SLT) oz. samodejno luščenje terminov je naloga obdelave naravnega jezika (ONJ), ki identificira specializirano terminologijo v domenskih korpusih. SLT se ne uporablja le pri terminografskih nalogah (npr. ustvarjanje specializiranih slovarjev), temveč omogoča tudi izboljšavo več drugih kompleksnih nalog s področja ONJ (npr. strojno prevajanje in luščenje informacij). Kljub temu, da je bil v zadnjih štiridesetih letih dosežen pomemben napredek pri samodejnem luščenju terminov, na področju SLT še vedno obstajajo pomembni izzivi.

V začetku naših raziskav se je večina sistemov za SLT zanašala bodisi na tehnike plitvega strojnega učenja bodisi na globoke nevronske mreže. Medtem ko so starejše tehnike trpele zaradi zamudnega postopka izdelave značilk in težav pri posploševanju na nove domene, so kasnejši pristopi modelirali nalogo kot problem binarne klasifikacije zaporedij z uporabo modelov arhitekture transformer. Vendar pa so imeli tudi ti začetni pristopi težavo, saj predstavlja generiranje vseh možnih n-gramov iz stavkov vseh dokumentov za namene učenja modela računske izzive. Poleg tega pa se novejši sistemi osredotočajo predvsem na nadzorovano učenje z zadostnim številom podatkov ter zanemarjajo gnezdene termine. V doktorski disertaciji zato predlagamo metode, s katerimi naslovimo te vrzeli.

V doktorski nalogi se osredotočamo na naslednje vidike. Prvič, v scenarijih, kjer obstajajo kvalitetne ročno označene učne množice za pristope nadzorovanega učenja, predlagamo izboljšave nevronskih pristopov z modeliranjem SLT kot problema klasifikacije žetonov (t. i. označevanje zaporedij). Tu kot osnovo uporabljamo modele z arhitekturo transformer, ki jim kot vhod dodajamo dodatne reprezentacije (npr. semantične reprezentacije oznak), ali pa spreminjamo dele arhitekture (npr. dodajanje dodatnih slojev na podlagi mešanice strokovnjakov (ang. mixture of experts) in rekurzivnih slojev). Poleg tega predlagamo nov sistem označevanja NOBI, ki omogoča boljše zajemanje gnezdenih terminov. Drugič, v scenarijih, kjer imamo omejeno število ročno označenih podatkov za učenje iz ciljnega jezika, podatki iz drugih jezikov pa so ustrezni za učenje modelov z nadzorovanimi pristopi, predlagamo čezjezično in večjezično učenje, s poudarkom na prenosu znanja iz jezikov z veliko viri na jezike z manj viri. Tretjič, v scenarijih, kjer imamo manj računskih virov ter podatkov ni dovolj za nadzorovane pristope, predlagamo nov pristop nenadzorovanega učenja, ki ga imenujemo *LlamATE* in temelji na velikih jezikovnih modelih. Sistem je sposoben luščenja terminologije s pomočjo znotraj-kontekstnega učenja (ang. in-context learning), kjer modelu kot vhod podamo le par primerov primernega luščenja terminologije, ki naj ga posnema pri luščenju terminologije iz vhodnega teksta. Sistem nato izboljšamo tudi s pomočjo tehnike samo-preverjanja (ang. self-verification).

Naša študija je prišla do naslednjih zaključkov. Prvič, modeliranje SLT kot klasifikacije žetonov (npr. z uporabo modela XLMR) je uspešen pristop, saj vodi do boljših rezultatov kot uporaba binarnih klasifikatorjev za klasifikacijo zaporedij ter hkrati potrebuje manj računskih kapacitet. Dodajanje sloja mešanice strokovnjakov na vrh globokega nevronskega modela (npr. (m)DeBERTA) dosledno izboljša kvaliteto modela v primerjavi z osnovnim modelom z navadnim linearnim slojem za klasifikacijo žetonov. Uporaba novega anotacij-

skega režima NOBI za učenje modelov za klasifikacijo žetonov, naučenih na dovolj velikem naboru podatkov z označenimi gnezdenemi termini, izboljša luščenje. Drugič, ko imamo na voljo manj podatkov, so pristopi klasifikacije žetonov primerni za prenos znanja iz drugih jezikov in domen, kar je pomembno predvsem pri prenosu na jezike z manj viri. Nazadnje, z našim pristopom, poimenovanim *LlamATE*, nakažemo potencial velikih jezikovnih modelov (LLM) za SLT, saj lahko uspešno opravijo nalogo s pomočjo samo nekaj podanih primerov luščenja terminov ter tehnike samopreverjanja. Ta pristop deluje tudi brez eksplicitnega poimenovanja domene ter pokaže na prenos znanja iz jezikov, ki so pri gradnji jezikovnih modelov dobro zastopani (npr. angleščina) na manj zastopane jezike. Čeprav ti modeli ne dajejo dovolj kvalitetnih rezultatov, da bi popolnoma nadomestili modele z nadzorovanim učenjem na ročno označenih podatkih, predstavljajo rešitve, ki ne potrebujejo ročno označenih podatkov, omogočajo pa vseeno dokaj dobro kvaliteto rezultatov.
Ključne besede: avtomatsko luščenje terminologije, arhitektura transformer, klasifikacija žetonov, veliki jezikovni modeli, oblikovanje in razvoj pozivov, učenje v kontekstu, Llama2, ChatGPT.

Contents

List of Figures

List of Tables

Abbreviations

ACTER	...	The Annotated Corpora for Term Extraction Research
ATE	...	Automatic Terminology Extraction
BERT	...	Bidirectional Encoder Representations from Transformers
BIO	...	Beginning-Inside-Outside (tagging)
BiLSTM	...	Bidirectional Long-Short-Term-Memory
CNN	...	Convolutional Neural Network
CRF	...	Conditional Random Field
DL	...	Deep Learning
DT	...	Decision Tree
HF	...	Human Feedback
GPU	...	Graphics Processing Unit
IAA	...	Inter-annotator Agreement
ICL	...	In-context Learning
IR	...	Information Retrieval
ISO	...	International Organization for Standardization
ML	...	Machine Learning
LIT	...	Labeled Semantics Information
LoRA	...	Low-Rank Adaptation
LLMs	...	Large-scale Language Models
LSTM	...	Long-Short-Term-Memory
MOE	...	Mixture-of-Expert
NE	...	Named Entitys
NER	...	Named Entity Recognition
NLP	...	Natural Language Processing
NMF	...	Non-negative Matrix Factorization
NOBI	...	Nested-Outside-Beginning-Inside (tagging)
OF	...	Output Format
OOD	...	Out-of-Domain
NMT	...	Neural Machine Translation
PBSMT	...	Phrase-Based Statistical Machine Translation
PoS	...	Part-of-Speech
PTBA	...	Phrase-Table-Based Alignment
RAG	...	Retrieval Augmented Generation
PTM	...	Probabilistic Topic Modeling
RoBERTa	...	Robustly Optimized BERT Approach
RL	...	Reinforcement Learning
RLHF	...	Reinforcement Learning with Human Feedback
RNN	...	Recurrent Neural Networks
RSDO5	...	Slovenian Corpus of Term-annotated Texts
SOTA	...	State-of-the-art
SGD	...	Stochastic Gradient Descent

SVM ... Support Vector Machine
TEI ... Text Encoding Initiative
XLMR ... Cross-Lingual Language Model With RoBERTa Architecture

Chapter 1

Introduction

In this section, we introduce automatic terminology extraction by first providing an overview of terms and terminology definitions and focusing on the challenges that their properties pose for terminology extraction. We then present various applications for terminology extraction and explain our motivation to overcome the existing challenges. We then state our hypotheses and give an overview of our methods to fulfill the hypotheses and solve the challenges we encounter in this task. We then explain our scientific contributions and the context of our research. Finally, we give an overview of the structure of the dissertation.

1.1 Automatic Terminology Extraction

Automatic terminology extraction or automatic term extraction (ATE) is a natural language processing (NLP) task that identifies specialized terminology from domain-specific corpora (see examples of terminology extraction results in Figure 1.1). The International Organization for Standardization (ISO) defines the process of term extraction as: " *terminology work that involves the identification and excerption of terminological data by searching through a text corpus*" (ISO:1087, 2019), where terminology work refers to work concerned with the systematic collection, description, processing and presentation of concepts[1] and their designations[2], terminological data is the data related to concepts and their designations, and text corpus represents a collection of natural language [3] data.

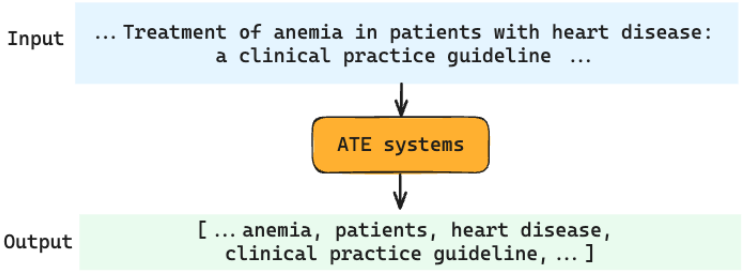


Figure 1.1: An example of an ATE system output from a given input sentence.

While ATE has traditionally been valuable and frequently used for several complex

[1]Unit of knowledge created by a unique combination of characteristics.

[2]Representation of a concept by a sign which denotes it in a domain or subject.

[3]Language in active use in a community of people, and the rules of which are mainly deduced from usage.

downstream tasks (e.g., machine translation (Wolf et al., 2011), information retrieval (Lingpeng et al., 2005), and sentiment analysis (Pavlopoulos & Androutsopoulos, 2014)), the advent of large language models (LLMs) has significantly altered its relevance in these areas. LLMs have advanced to the point where they can handle these tasks with high accuracy without relying heavily on pre-extracted terms. As a result, the utility of ATE in these domains has diminished. However, ATE remains crucial for specific applications within terminology management, particularly in the creation and maintenance of specialized glossaries (Maldonado & Lewis, 2016) and dictionaries (Le Serrec et al., 2010). In these contexts, ATE continues to provide efficient and accurate extraction of domain-specific terms, facilitating precise communication and knowledge management within specialized fields. Despite the shift brought by LLMs, ATE still plays a vital role in ensuring consistency and accuracy in terminology management. In recent years, great progress has been made in the automatic recognition and classification of term candidates from specialized corpora. Nevertheless, ATE still represents a major challenge.

1.1.1 Terms and Terminology

We dedicate the first part of the introduction to defining the terms and terminology as these concepts form the core of our dissertation. The definitions outlined by the International Organization for Standardization (ISO) serve as a starting point from which the *"Terminology work and terminology science — Vocabulary"*[4] (ISO:1087, 2019) provides a systematic description of the concepts related to terminology work and terminology science and to clarify the use of the terms in this field. The following two definitions are directly taken from ISO:1087 (2019):

Terminology: *set of designations and concepts belonging to one domain or subject,*
Term: *designation that represents a general concept by linguistic means,*

where designation is defined as the representation of a concept by a sign that denotes it in a domain or subject; concepts refers to the unit of knowledge created by a unique combination of characteristics; general concepts represents a concept that corresponds to a potentially unlimited number of objects which form a group by reason of shared properties; domain is the field of special knowledge; and subject as an area of interest or expertise. As mentioned in the definition, the fundamental characteristic of terminology is its relation to a specific domain, which distinguishes it from the general language [5].

In *Glossary of Terms*, De Bessé et al. (1997) defined a *term* as a *"lexical unit consisting of one or more than one word which represents a concept inside a domain"*, and provided three different definitions for *terminology*, including: [1] *"The vocabulary of a subject field"*. The other two meanings of *terminology* from De Bessé et al. (1997) are related to the [2] *"The study of terms, concepts, and their relationships"*; and [3] *"The set of practices and methods used for the collection, description, and presentation of terms"*. While the first definition of *terminology* is similar to ISO's, the other two relate more to the activities and studies of terms. Kageura (2012) defined the inter-relation between term and terminology as follows: *"...we need the concept "terminology" to pursue the study of terms and terminologies, while a concrete study should start from individual terms or a set of terms that is regarded as representing a terminology..."*.

[4] https://www.iso.org/obp/ui/en/#iso:std:iso:1087:ed-2:v1:en

[5] General language is characterized by the use of linguistic means of expression independent of any specific domain.

Similar to De Bessé et al. (1997), several researchers and experts have investigated the classical definition of *terms*, i.e., *"any conventional symbol representing a concept defined in a subject field"* (Felber, 1984); *"words that are assigned to concepts used in the special languages that occur in subject fields or domain-related texts"* (Wright, 1997); or *"lexical units used in a more or less specialized way in a domain"* (Kageura, 2012). Calling *terms "lexical units"* instead of *"words"* better highlighted two main characteristics: [1] Terms can be both single-word (e.g., *"acetylcholinesterase"*) and multi-word (e.g., *"adenylyl cyclase"*, *"amino-terminal pro-b-type natriuretic peptide"*); and [2] the relation of the term to a specialized domain or subject. Cabré Castellví and Sager (1999) specified the later characteristic as follows: *"...most salient distinguishing feature of terminology in comparison with the general language lexicon lies in the fact that it is used to designate concepts about special disciplines and activities..."*. Conversely, the modern approaches considered terms to be *"linguistic units that are subject to all of the same phenomena as general lexica, including variation and ambiguity"* (L'Homme, 2020). However, defining a boundary between a specialized language and a general one is still ambiguous. Kageura (2015) concluded that *"what makes some lexical unit terms is their usage and social recognition within a given domain, subject or vocation. Without such extra-linguistic information, we cannot identify lexical units in a given text as "terms""*. This partially corresponds to previous studies, stating that *"...formally terms are indistinguishable from words..."* (Sager, 1998) and *"...There is no fully operational definition of terms..."* (Gaussier, 2001).

The idea of these definitions was shared by various later versions of ISO until the *"Health informatics — Clinical particulars — Core principles for the harmonization of therapeutic indications terms and identifiers"*[6] (ISO/TS 5499:2024), in which *terminology* and *term* are largely interchangeable.

Terminology: *structured, human readable and machine-readable representation of concepts,*
Term: *linguistic representation of a concept,*

Term and *terminology* can be used interchangeably in some contexts. For instance, *automatic term extraction* and *automatic terminology extraction* refer to the same task, and are sometimes even used within the same text. This dissertation will use *term* and *terminology* interchangeably, as while we extract individual designations, the final result is a set of the candidate terms, i.e., the terminology.

Despite the involvement of domain experts and specialists in the naming of scientific concepts, terminology was not considered a scientific discipline or field of study until the work of Wüster (1974, 1991), in particular, his general theory of terminology was an inspiration for terminology research studies. In Wüster (1974), he explained that this general theory of terminology spreads into other disciplines such as linguistics, logic, ontology, information science, and other sciences; a fact that makes terminology interdisciplinary (drawing from several fields of study) and multidisciplinary (making use of several disciplines at once). Inspired by previous research, Valeontis and Mantzari (2006) also described *terminology* as an interdisciplinary field: *"The interdisciplinarity of terminology results from the multifaceted character of terminological units, as linguistic items (linguistics), as conceptual elements (logic, ontology, cognitive sciences) and as vehicles of communication in both scientific and generic language contexts"*. Pimentel (2015) emphasized the *terminology* as a separate scientific discipline: *"grown into a multi-faceted science, which seems to have reached adulthood"*.

[6] https://www.iso.org/obp/ui/#iso:std:iso:ts:5499:ed-1:v1:en:term:3.1.11

1.1.2 Applications

The increasing interest in terminology is no surprise given the constantly growing volume of specialized communication. However, the manual extraction of the candidate terms is often time-consuming and labor-intensive. Therefore, ATE was born as an NLP task that reduces the effort of manually identifying terms from domain-specific corpora by providing a list of candidate terms. This section focuses on illustrating the importance of ATE tasks with application examples, which can be divided into three categories: [1] ATE for terminology management (Le Serrec et al., 2010; Ortego-Antón, 2021; Repar et al., 2019b), [2] ATE as part of an NLP pipeline (Terryn et al., 2019; Yan et al., 2017), and [3] ATE as a commercial product (e.g., TERMite, Sketch Engine).

1.1.2.1 Automatic Terminology Extraction for Terminology Management

Maldonado and Lewis (2016) considered ATE the first step in many terminology management processes (e.g., glossary construction and curation). Le Serrec et al. (2010) investigated how standalone and combined single-word terminology extraction and bilingual lexical alignment assisted terminologists when compiling bilingual English-French dictionaries on climate change. The French candidate terms were first extracted using TermoStat (Drouin, 2003) and then submitted to the lexical aligner named Alinea. The results of lexical alignment were analyzed based on a typology of terminology equivalents, which proved the valuable potential of both tools to compile bilingual dictionaries. Similarly, Ortego-Antón (2021) took advantage of the task to create a Spanish-English specialized dictionary about dried meats with the following procedure: Given the comparable non-aligned corpus as the input, TermoStat (Drouin, 2003) extracted the candidate terms using L'homme (2004)'s principles for terminography, thus, building a bilingual termbase with the specialized tool MultiTerm[7] and finding all relevant information with a freeware tool for corpus analysis named AntConc[8]. Repar et al. (2019b) introduced TermEnsembler, a system for semi-automated bilingual terminology extraction and alignment, focusing on English and Slovenian. TermEnsembler proposed a novel Phrase-Table-Based Alignment (PTBA) method based on Pialign (Neubig et al., 2011) and a new method using an evolutionary algorithm to combine solutions of an ensemble of elementary term alignment algorithms. The system developed for one of Southeast Europe's largest language service providers, achieved up to 96% accuracy in aligning terms in the 400 best-matching term pairs.

Recently, the Slovenian Terminology Portal[9], developed as part of the Development of Slovene in a Digital Environment Project, is a platform which allows registered users to create their specialized corpus, analyze it with an online concordance tool, use a terminology extraction tool to extract candidate terms from their texts or already published texts, and import the list of extracted candidate terms into the editor where they can add definitions or equivalents in foreign languages to create their terminology resource. The terminology extraction process is based on statistical and neural methods, including the neural sequence labeling approach (H. Tran et al., 2022) developed as part of this thesis. This is one of the significant contributions to smaller lesser-known European languages such as Slovenian and one of the first tools that leverage deep learning (transformer-based models) as the base model for terminology extraction tasks.

[7] https://www.trados.com/product/multiterm/
[8] https://www.laurenceanthony.net/software.html
[9] https://terminoloski.slovenscina.eu/

1.1.2.2 Automatic Terminology Extraction as Part of an NLP Pipeline

ATE had served as a powerful building block within an NLP pipeline by acting as a preprocessing step or an information distillation tool, which can enrich various downstream tasks, such as information retrieval (Lingpeng et al., 2005), machine translation (Wolf et al., 2011), aspect-based sentiment analysis (Pavlopoulos & Androutsopoulos, 2014), hypernym detection (Rigouts Terryn et al., 2016), and search engine optimization (Terryn et al., 2019). However, with the advent of LLMs, this direction became less important but identifying technical terminology and aligning them is still a challenge. No systematic evaluation of the terminology extraction on other downstream tasks (e.g., machine translation), leaving an open discussion about the impact of LLMs with and without additional information from specialized terminology dictionaries.

Information Retrieval (IR). ATE tasks have been commonly used for re-ordering steps in IR systems to improve document-level precision. L. Yang et al. (2004) used the extracted long terms in query and documents to reorder retrieved documents in Chinese IR systems with a four-step procedure: [1] cluster the whole document set into clusters, [2] extract global key terms from these clusters, [3] find local terms in a query or a document using these global terms and their frequencies, and [4] re-calculate the similarity between query and document using long local terms and re-order the retrieved documents using the new similarity value. Likewise, Lingpeng et al. (2005) extracted the candidate terms in the query and calculated their importance. They then re-ordered retrieved documents by the kinds of terms in the query they contain to boost the precision of Chinese IR tasks.

Machine Translation. An important part of translating domain-specific terms, which covers a large portion of the task (Katan, 2009), is finding the correct equivalents for domain-specific terms. Fähndrich (2005) estimated that around 20-50% of translators' working time was spent on terminology research. Bowker (2015) has proven that a well-managed terminology system could improve the quality of a translation, reduce the time and cost of the process, improve corporate branding, and prevent legal liabilities. An example of a significant contribution to the translation task was provided by Xiong et al. (2016). It has integrated three models (e.g., for solving term translation disambiguation, term translation consistency, and term unithood, respectively) into a hierarchical phrase-based statistical machine translation system to improve the accuracy of translation of domain-specific terms. Haque et al. (2018) proposed TermFinder, a bilingual multi-word term extractor that uses log-likelihood comparison to extract source and target terms independently from both sides of a parallel domain corpus, and align them using the Phrase-Based Statistical Machine Translation (PB-SMT) model (Koehn et al., 2003). Haque et al. (2020) facilitated the assessment of terminology errors in machine translation by using PB-SMT in combination with neural machine translation (NMT). This has improved English-Hindi and Hindi-English translation and refined the quality of terminology translation.

Aspect-based Sentiment Analysis. ATE systems have also proven their usefulness in aspect-based sentiment analysis where the polarity can be analyzed for specific targets rather than for an entire sentence or text, as in sentiment analysis. De Clercq et al. (2015) applied the TExSIS (Macken et al., 2013) term extractor to identify relevant entities for aspect-based sentiment analysis in restaurant and laptop reviews. The identified entities are then aggregated into broader aspect categories and fed into a lexical and pointwise mutual information-based classifier to learn and infer how strongly an aspect is associated with positive or negative sentiments. Similarly, a comparable approach has been proposed for Russian (Mayorov et al., 2015).

Hypernym Detection. Rigouts Terryn et al. (2016) extracted the candidate terms for the automatic recognition of Dutch hypernyms. The recognition was a three-stage procedure consisting of three consecutive modules: [1] a pattern-based regular expression module, [2]

a morphosyntactic-based hypernym recognition module, and [3] a complex noun decomposition module. The ATE task was used to find relevant single and multi-word candidate terms for which hypernyms should be recognized, and as an additional filter to remove candidates with an unlikely part-of-speech (PoS) pattern unless they received a high termhood score.

Search Engine Optimization. Terryn et al. (2019) took advantage of TExSIS (Macken et al., 2013), a hybrid tool for bilingual ATE from parallel corpora, to improve the search engine on a medical website. While this strategy was more error-prone and required more extensive manual validation, it was able to lead users to relevant information that did not necessarily contain their exact search terms.

1.1.2.3 Automatic Terminology Extraction as Commercial Products

This section explores commercially available software tools developed for automatic terminology extraction. These systems process text data to identify key terms from text data in a variety of domains (e.g. patent documents and medical reports).

TERMite[10] (TERM identification, tagging & extraction tool), developed under SciBite (an ELSEVIER company), is the paid, ultra-fast extraction engine for semantic analysis. TERMite took advantage of high-quality, hand-curated vocabularies and ontologies (VOCabs), such as MeSH[11], Uniprot[12], or MedDRA[13], to extract candidate terms from scientific medical texts, transforming unstructured content into rich, machine-readable data.

NoSketchEngine[14] and Sketch Engine[15] software tools explore how language works with 800 ready-to-use corpora in more than 100 languages. Each corpus contains up to 80 billion words in size to provide a truly representative sample of the language. Both versions include terminology extraction and bilingual terminology extraction with combined statistical and linguistic analysis. While the former extracts the candidate terms in the documents or in specialized texts that the user uploads or that the Sketch Engine can find on the Internet, the latter is carried out using the translation memory that the user uploads. The result is a bilingual list of candidate terms and their translations. The extracted candidate terms require very little, if any, manual cleaning or post-processing.

Tilde Terminology[16] is a cloud-based tool that uses a combination of linguistic analysis and statistical methods to recognize and extract term candidate terms. Tilde investigates a variety of features, including the ability to search for terms in the largest European terminology database, create custom and shared term collections in a simple cloud-based terminology platform, and filter terms by PoS, frequency, and other criteria.

MultiTerm Extract[17] is an application that checks the frequency of terms at a subsegment level and enables the user to build project glossaries without having to manually search for the terms. The main purpose of MultiTerm is to automatically identify and extract the candidate's monolingual or bilingual terms from existing documents. MultiTerm provides a central place to store and manage multilingual terminology, helping to quickly create termbases and glossaries and deliver consistent, high-quality content from source to translation.

Wordbee[18], as part of the Wordbee Translation Suite, is an all-in-one terminology so-

[10] https://scibite.com/platform/termite/
[11] https://www.ncbi.nlm.nih.gov/mesh/
[12] https://www.uniprot.org/
[13] https://www.meddra.org/
[14] https://www.sketchengine.eu/nosketch-engine/
[15] https://www.sketchengine.eu/
[16] https://term.tilde.com/
[17] https://appstore.rws.com/plugin/109
[18] https://wordbee.com/termextractor/

lution specifically designed for translators to extract candidate terms, related synonyms and variants from bilingual documents. The terminology assets follow the TermBase eXchange standard (TBX 3.0), an open XML-based international standard for the exchange of structured terminological data that makes it easy for users to identify, filter, and export terms from project files.

1.1.3 Challenges

Despite the importance of term identification and the great attention that this task receives in research, correct recognition and extraction of terminology remain a difficult task due to the following challenges.

1.1.3.1 Consensus on the Definition of a Term

Despite several different definitions (as discussed in Section 1.1.1) that have been proposed to describe the meaning of a term, and despite efforts to distinguish between terms and general vocabularies (Hätty & im Walde, 2018; Hoffmann, 1985; Pearson, 1998), there is still a lack of a clear and consensual distinction between terms and non-terms (Pazienza et al., 2005).

The definition of *term* is still arbitrary and controversial. It is difficult to agree on the difference between terms and general vocabulary in a corpus since terms vary between different domains, languages, and applications (Hoffmann, 1985). The terms may differ significantly in different domains, which makes it difficult to develop universal methods for terminology extraction. For example, as emphasized in the work of De Bessé et al. (1997), a *term* can be defined as *"a lexical unit consisting of one or more than one word which represents a concept inside a domain."*. In the context of technical domains, De Bessé et al. (1997) classified terms into three categories: [1] general terms of a subject field (e.g., descriptions, instructions and textbooks, patent descriptions, and other non-industry-specific terms), [2] craft-, industry-specific- and even firm-specific terms, and [3] product-specific terms. Meanwhile, in relation to applications, Hätty and im Walde (2018) discussed that the definition of terms is much more relevant for annotators (e.g., annotating terms for different applications) and applications (e.g. different professionals identifying the candidate terms).

The disagreement in term definition also extends to various aspects. L'Homme (2020) addressed this problem with three questions: [1] whether the lexical unit is relevant to the subject field, e.g., the length of the terms (single and/or complex terms), [2] whether proper nouns can be terms (e.g., PoS patterns), and [3] what is the level of terminological variation (e.g., whether or not named entities should be included as part of the terms).

On the one hand, Bourigault (1992) emphasized the difficulty of extracting one-word terms: *"term extractors focus on multi-word terms for ontological motivations: single-word terms are too polysemous and too generic and therefore it is necessary to provide the user with multi-word terms that represent finer concepts in a domain"*. Meanwhile, Heylen and De Hertog (2015) explained that this difficulty was due to the lack of good statistical measures to evaluate, for example, *termhood* and *unithood*. Formally, *unithood* refers to "*the degree of strength or stability of syntagmatic combinations and collocations*" (Kageura & Umino, 1996), and termhood is defined as "*the degree that a linguistic unit is related to domain-specific concepts*" (Kageura & Umino, 1996). Although the former is only relevant to complex terms, the latter concerns both simple terms and complex terms. Recently, termhood has often been equated with unithood, focusing on complex and multi-word terms. As termhood measures are now more readily available and applicable thanks to

increased computational power and the availability of large corpora, some approaches take the opposite approach and focus exclusively on one-word terms (Nokel et al., 2012).

On the other hand, L'Homme (2020) has emphasized PoS factors as follows: *"The vast majority of terms are nouns. This is a consequence of the focus of terminology on concepts (most of them are entities) and the way concept is approached. Even in cases where activity concepts (linguistically expressed by nouns or verbs) or property concepts (prototypically expressed by adjectives) need to be taken into account, nouns are still preferred".* Recently, Rigouts Terryn et al. (2021) raised a further question as to what types of entities can be included as terms and emphasized the need for consistency, that all extracted terms must be specific to the domain of the analyzed text, whether with or without named entities.

In short, there are still no clear and fully agreed formal characteristics that distinguish terms from general language. Criteria such as the length and PoS of terms, the inclusion of proper names, and relevance to a particular domain all depend on the context in which they are to be employed.

1.1.3.2 Data Acquisition Bottleneck

Astrakhantsev et al. (2015) dealt with the bottleneck in data collection as a challenge in terminology extraction tasks, especially in the creation of the gold standard. Building domain-specific corpora with manual terminology annotation is error-prone, labor-intensive, time-consuming, and subjective. There are two main reasons for this: the notoriously low level of agreement between annotators and the lack of consensus on an annotation protocol. As a result, it is pragmatically problematic to combine multiple datasets into a diverse and comprehensive corpus. Even in the most transparently annotated corpora such as ACTER (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020), the current benchmark dataset, although parts of the corpora were annotated by multiple annotators to calculate inter-annotator agreement, while language students helped with the annotation, the bulk of the annotation work was done by a single annotator. As terminology is highly subjective, this dataset is inevitably partially subjective. Building a corpus that fulfills the requirements of transparency and subjectivity (e.g., through semi-automatic checks, the development of guidelines, and agreements between annotators) is essential.

The difficulty of obtaining parallel corpora for specific domains is also an obstacle to multilingual terminology extraction. One solution is to switch to comparable corpora instead (Delpech et al., 2012; Rigouts Terryn, Hoste, & Lefever, 2020). The corpora then contain texts in different languages that are not translations but have almost the same vocabulary on a comparable topic. In contrast to parallel corpora, however, it is impossible to know where these equivalent terms can be found or whether they exist at all.

1.1.4 Motivation

The field of terminology extraction has come a long way, but there is still room for advancement. In this section, we highlight the current landscape, identify the gaps and a range of novel methods developed in the scope of the thesis to fill them.

1.1.4.1 Terminology Extraction Improvement from the Perspective of Sequence-Labeling Models

Our goal was to improve the performance of extracting the candidate terms from text sequences by considering the task as a token classification task using different transformer variants.

Thanks to the TermEval 2020[19] shared task, several deep learning (DL) based methods were proposed to solve the challenges. However, the winning solutions (Hazem et al., 2020) and the latest research (Lang et al., 2021) mostly considered the ATE tasks as binary sequence classification tasks. These approaches relied on the positive (is-a-term) and negative (not-a-term) samples generated from all possible n-grams of a fixed length of a given sentence. Despite the superior predictive performance compared to other ML-based approaches, the process of generating all possible n-grams from each sentence across all documents for training purposes poses computational and storage challenges.

Meanwhile, the latest trend is the use of token classification methods where the candidate terms are detected in their original contexts, usually by classifying each token in the text sequence as (part of) a term or not. Thus, instead of having a label for the whole span of the text sequence, the token classifier assigns a label for each token. Nonetheless, a majority of the works tested this mechanism on non-neural models (e.g., CRF (Judea et al., 2014; Kucza et al., 2018)) or classical neural ones (e.g., LSTM, BiLSTM, LSTM-CRF, and BiLSTM-CRF (Andrius, 2020; Han et al., 2018; Hazem et al., 2022; N. T. Le & Sadat, 2021)).

With the advent of language models, we embraced transformer-based token classification as an alternative strategy of binary sequence classification for ATE tasks. On the one hand, since this approach operates without upfront n-gram generation and handles each sentence as a single example that needs to be labeled, it is considerably more time-efficient than the previous binary sequence classification approach. On the other hand, transformer variants have several advantages over classical neural approaches such as parallelization (e.g., analyzing all relationships between words simultaneously) and better contextual understanding (i.e., a deeper understanding of context and relationships between words thanks to self-attention mechanisms).

1.1.4.2 Terminology Extraction Improvement from the Perspective of Annotation Regimes

Another aspect that is important for terminology extraction is the recognition of nested terms. Nested terms are defined as "*terms that appear within other longer terms, and may or may not appear by themselves in the corpus*" (K. Frantzi et al., 2000). We have dived deeper into improving the extraction of candidate terms by developing a new annotation procedure to improve the efficiency of the classifier in capturing nested terms.

In many practical applications, it is common for terms to have a nested structure, where one term can contain other terms or be part of others. Š. Vintar (2004) initially proposed to rank and/or discard nested terms based on the C-value, but the results of the proposed approach were not satisfactory. Marciniak and Mykowiecka (2015) later identified them by combining grammatical correctness and normalized pointwise mutual information (NPMI) based on bigrams in a corpus. However, the efficiency of this method strongly depends on the corpus features (e.g., size, thematic homogeneity, and phrase frequency). Recently, Gao and Yuan (2019) proposed an end-to-end architecture that formulated the task as the progress of classifications and ranking, which considered all possible n-gram spans or segmentation in the sentence and distinguished whether they can be domain-specified terms in the sentence. Nonetheless, this suffers from reduced recall due to ranking and its threshold output does not apply to new, unseen domains.

For other downstream NLP tasks that share the same mechanisms (e.g., named entity recognition, keyword extraction), there are other approaches in addition to the common

[19] A platform for researchers working on ATE systems with the ACTER corpora (https://termeval.ugent.be/).

sequence labeling tag regimes (e.g., BIO (Ramshaw & Marcus, 1999), IOBES (Lester, 2020), BMEWO (Ratinov & Roth, 2009), BILOU (Ratinov & Roth, 2009)). However, none of them, except the BIO regime for the sequence-labeling approach, has been applied yet for terminology extraction.

Furthermore, our study (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022) pointed out that the two most common errors made by the tested classifiers were to predict a shorter term nested in the ground truth term and vice versa, i.e., the model sometimes generated the terms not covered in the ground truth, containing a nested term. This insight led us to a hypothesis about the insufficiency of the widely used BIO labeling regime (Hazem et al., 2020). This regime does not allow labeling the nested terms and giving the model the necessary information to avoid the above mistakes. Therefore, our goal has been to develop a new annotation regime to better capture nested terms to improve the classifier's performance.

1.1.4.3 Improvement via Cross-domain, Cross-lingual, and Multilingual Learning

Our goal was to bridge the gap in techniques between extracting general information and domain-specific terms. We developed different approaches that allow modeling and accounting for the domains with specialized language (including dominant and lesser-known European languages) while benefiting from the knowledge gained from common languages and vocabularies.

Current research on ATE tasks deals with two major issues: [1] the data collection for a specific field of expertise or domain is often non-existent or modest, and [2] the available well-annotated corpora are scarce, especially for lesser-known languages (e.g., Slavic languages). To address the former issue, J. Liu et al. (2018) proposed to use a concatenated vector of the one trained on the specialized corpora and the general corpora as a new word embedding. In other words, they concatenated a relatively small size word vector from the specialized corpora and a relatively large size one from the general corpora to increase the total training corpora. However, this technique is difficult to apply to most ATE tasks, as it still requires an "appropriate" amount of domain-specific, manually annotated data, which is scarce (i.e., the second problem).

In addition, zero-shot and few-shot learning have gained popularity as advanced approaches to overcome the problem of limited availability of annotated corpora. In zero-shot learning, a model must categorize examples or instances into classes without seeing the examples of those classes during training. In contrast, few-shot learning allows the model to categorize examples or instances into classes with only a handful of training examples. In our case, both are based on the transfer of knowledge that the model receives from a rich domain/language with lots of annotated data to a lesser domain/language with little or no data.

Therefore, we proposed three settings: [1] cross-domain learning throughout our investigation, [2] cross-lingual learning, and [3] multilingual learning. The cross-domain learning checks the generalization capabilities (and thus usefulness) of the model, i.e., how successfully the knowledge the model obtains on one domain can be applied to an arbitrary new unseen domain (i.e., zero-shot learning at the domain level). Meanwhile, cross-lingual learning examines how well the ATE model performs without the language-specific training corpus and how good the knowledge transfer between different languages is (i.e., zero-shot learning at the language level). In addition, we investigated whether adding more data from other languages to the training set in the target language improves the model's performance through multilingual learning.

1.1.4.4 Terminology Extraction Improvement from the Perspective of Generative Models

The emergence of large-scale language models (LLMs) has significantly improved performance on several downstream tasks (Vilar et al., 2022). Two strategies for integrating LLMs in these tasks include fine-tuning and in-context learning (ICL) techniques. While the fine-tuning involves initializing a pre-trained model and conducting additional training epochs on task-specific supervised data, ICL leverages the ability of LLMs to generate texts with only a few task-specific examples as demonstrations. The concept of prompts with few-shot demonstrations was first introduced by Radford et al. (2019), followed by an empirical analysis of the ICL paradigm with GPT-3 (Brown et al., 2020) and PaLM (Chowdhery et al., 2022). With the release of ChatGPT[20] by OpenAI and the blossoming of open-source LLMs (e.g., Llama (Touvron et al., 2023), Mistral (Jiang et al., 2023), Zephyr (Tunstall et al., 2023)), recent research has focused on evaluating their performance in various NLP tasks. Despite their attraction, none of these systems has yet been used to solve the task, leaving a gap in terminology research.

We examined how well LLMs were able to extract the domain-specific terms and compared their performance through an empirical study of different prompting strategies across languages. We bridged the gap between text generation and sequence labeling inherent in the ATE task by guiding LLMs to produce predictions with specific output formats. Instead of fine-tuning directly, we applied the ICL technique, where we considered LLMs as predictors and prompted the model with a few examples of sentences covering terms and non-terms in mono- and cross-domain/lingual transfer. In this way, we evaluated their potential use within pragmatic, real-world applications characterized by the limited availability of labeled examples.

1.2 Hypotheses and Methodology Overview

In this dissertation, we formulate three primary directions on how we approach the task of terminology extraction with respect to methodology and annotation regimes, namely [1] *Terminology Extraction from the Perspective of Sequence Labeling Models*; [2] *Terminology Extraction from Perspective of Annotation Regimes*; and [3] *Terminology Extraction from the Perspective of Generative Models*. We present our hypotheses for each direction and briefly describe how we formulated and verified them.

1.2.1 Terminology Extraction from the Perspective of Sequence-Labeling Models

In this direction, we have identified five different hypotheses from the perspective of sequence-labeling models, which we evaluate in Chapter 4 of this dissertation.

1.2.1.1 H1: Terminology Extraction Benefits from Sequence Labeling Models

[H1.1] Token Classification Models vs. Binary Classification Models: *"A token classifier trained on a monolingual dataset in cross-domain setting surpasses the performance of binary classification system in extracting the candidate terms."*

We considered terminology extraction as a token classification (so-called sequence labeling) task and the transformer-based (e.g., XLMR) models as a token classifier where the model returned a label for each token in a text sequence using the BIO labeling scheme. We

[20] https://openai.com/chatgpt

compared our proposed approaches with the benchmarks that used ML- and transformer-based models as binary classifiers to verify our hypothesis. Both models were implemented in cross-domain settings where the classifier predicts new unseen domains.

[H1.2] Cross-lingual Transfer vs. Monolingual Learning: *"In a zero-shot cross-lingual setting, a token classifier achieves comparable results to monolingual training in a target language."*

We evaluated the capability of the model to apply the knowledge learned in one or more languages for ATE in another unseen language when no data was available for a target language. Therefore, we fine-tuned the terminology extraction model in one or more languages (e.g., English and Dutch) and tested it in another language, not appearing in the train set (e.g., French). To prove our hypothesis, we compared our proposed approaches with the monolingual setup where the model performs when there is a language-specific training corpus available and there is a match between the language of the train set and that of the test set.

[H1.3] Multilingual Learning vs. Monolingual Learning: *"A token classifier trained on multilingual datasets and applied to a seen target language outperforms the monolingual models trained on the target language and cross-lingual models not trained on the target language."*

We investigated whether adding more data from other languages to the training set in the target language improves the predictive performance by fine-tuning models using [1] training datasets from those languages in the ACTER (English, French, and Dutch) or [2] training datasets from the languages in the ACTER plus the Slovenian training dataset from the RSDO5 corpus, and then apply the model to the test sets of all languages in the ACTER dataset. To prove our hypothesis, we compared our proposed approaches with monolingual and cross-lingual setups.

[H1.4] The Impact of Labeled Semantics Information in Terminology Extraction: *"The integration of label semantic information into a token classifier based on BERT outperforms the base model."*

We proposed LIT, a novel architecture consisting of four components: an Encoder, a Feature extractor, a Decoder, and a Cosine similarity operation. The first three components worked together to generate an embedding for the target token. This embedding was then compared to the embeddings of term labels generated by the decoder using cosine similarity. The final prediction was based on the resulting similarity score. We test the hypothesis in a monolingual cross-domain setting.

[H1.5] The Impact of Mixture of Experts in Terminology Extraction: *"A novel token classification head architecture that combines a mixture of experts (MoE) and recurrent neural networks (RNN) on a transformer-based model outperforms the base token classification model."*

We proposed MOSES, a novel architecture with (m)DeBERTa as the backbone and a new token classification head containing different layers of the mixture of experts (MoE) and recurrent neural networks (RNN). We were interested in the specific contributions that the MoE and RNN layers have on the model's performance (either individually or together). Therefore, we compared the settings that contained these components with the baseline (m)DeBERTa model with a conventional (dense) token classification head.

1.2.2 Terminology Extraction Perspective of Annotation Regimes

Using sequence-labeling approaches, we have shown the obstacles in capturing the nested term and proposed a hypothesis from the perspective of annotation regimes, which we evaluate in Chapter 5 of this dissertation.

1.2.2.1 H2: Terminology Extraction Benefits from Nested Annotation Regime

[H2.1] The Impact of Nested Term Annotation in Terminology Extraction: *"An annotation regime that captures additional information with regard to nested terms, improves the performance of token-based terminology extraction."*

The error analysis revealed that the two most common errors that the token classifiers made with the standard BIO annotation were that they predicted a shorter term nested in the ground truth term (prediction: *mass spectrometric*, ground truth: *mass spectrometric analysis*) and vice versa, i.e., the model sometimes generated longer candidate terms not covered in the ground truth that contained a nested term (prediction: *epithelial to mesenchymal transition marker*, ground truth: *epithelial* and *mesenchymal*). This insight led to a hypothesis about the insufficiency of the widely used BIO labeling regime. This regime did not allow labeling of the nested terms and failed to provide the model with full information on nested terms to avoid the above mistakes. Therefore, we proposed a new NOBI annotation regime to capture single nested terms better and tested the hypothesis that the NOBI regime leads to higher results than the standard BIO regime.

1.2.3 Terminology Extraction from Perspective of Generative Models

We investigated another direction in which we took advantage of generative models for terminology extraction. In this direction, we formulated the problem into three hypotheses and evaluated them later in Chapter 6 of this dissertation.

1.2.3.1 H3: Terminology Extraction Benefits from Generative Models

[H3.1] Terminology Extraction as Seq2Seq Classification Tasks: *"Token classification model outperforms Seq2Seq models on terminology extraction task."*

We considered terminology extraction as a template Seq2Seq ranking task where the original sentence was the source sequence, and the template filled with the candidate term spans was the target sequence during training. To prove our hypothesis, we compared our approaches with the benchmark of sequence labeling (token classification) approach.

[H3.2] Large Language Models (LLMs) as Instructors for Terminology Extraction: *"Large-scale language models with few-shot demonstration prompting using generative output formats leads to slightly lower performance, but avoids the need for extensive data annotation."*

We have investigated the applicability of open-source (*LLama2-chat*) and closed-source (*ChatGPT*) large-scale language models (LLMs) in the ATE task in three prompting-output designs: [1] sequence-labeling response, [2] text-extractive response, and [3] filling in the gap between the two types by text-generative response. In doing so, we hypothesized that while prompting sacrificed some performance compared to fully supervised sequence-labeling baselines (a loss of a few percentage points), it offered a valuable trade-off by eliminating the need for extensive data annotation and computational efforts. This demonstrated the potential of our model for real-world applications characterized by the limited availability of labeled examples.

[H3.3] The Domain Is Important for Automatic Terminology Extraction in the Era of LLMs: [1] *"When employing LLMs for terminology extraction, few-shot demonstration prompting with self-verification allows us to predict terms without needing explicit information about the domain of the examples. This works for examples within the same domain as well as across different domains."*; [2] *"Using LLMs for few-shot demonstration prompting in cross-lingual transfer, with self-verification, allows for effective transferring of knowledge from well-represented languages to less-represented ones."*

We proposed *LlamATE*, a framework to verify the impact of domain specificity on ATE when using ICL prompts in open-sourced reinforcement learning with human feedback (RLHF) models, namely *Llama-2-Chat*. We evaluated how well instruction-tuned models perform with different levels of domain-related information in both rich-resourced (English and French) and lesser-resourced European languages (Dutch) from ACTER datasets, i.e., in-domain and cross-domain demonstrations with and without domain enunciation. Furthermore, we examined the potential of cross-lingual and cross-domain prompting to reduce the need for extensive data annotation of the same domain and language. In addition, we shed light on the influence of the self-verification step in the procedure to reduce the impact of noise and hallucination in the prediction. Last but not least, we aimed to evaluate the practical use of *LlamATE* for low-resourced terminology extraction tasks.

1.3 Scientific Contributions

In this section, we describe the main contributions of this dissertation and the resulting scientific findings, including our related publications (e.g., journals, conferences, and workshop papers).

1.3.1 Main Contributions

This dissertation aims to propose a new mechanism to extract candidate terms and nested terms more efficiently with the following contributions to the scientific community:

1. *Introducing terminology extraction as a transformer-based token classification (so-called sequence-labeling) task (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022; H. T. H. Tran, Martinc, Pelicon, et al., 2022; H. Tran et al., 2022; T. H. H. Tran et al., 2022).* Before the beginning of our dissertation, most of the existing methods were limited to rule-based systems (e.g., TermoStat (Drouin, 2003), TExSIS (Macken et al., 2013)), ML-based approaches (e.g., (Rigouts Terryn et al., 2021)) or binary sequence classification ones (e.g., (Hazem et al., 2020)). The use of neural networks for terminology extraction as a token classifier still fell behind the recent advances in language models. Before our works, most of the token classifiers applied to terminology tasks included either classical ones (e.g., LSTM, BiLSTM) or the vanilla BERT. This is one of the first studies to embrace the task as a token classification task using transformer-based language models (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022; H. T. H. Tran, Martinc, Pelicon, et al., 2022; H. Tran et al., 2022; T. H. H. Tran et al., 2022) with empirical studies on two aspects: [1] monolingual vs. multilingual pre-trained models; and [2] masked (e.g., BERT, RoBERTa) vs. autoregressive (e.g., XLNet) models. The source code can be found at https://github.com/honghanhh/terminology-extraction.

2. *Filling the research gap between rich- and lesser-resourced European languages (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022; H. T. H. Tran, Martinc, Pelicon, et al., 2022).* Due to the data acquisition bottleneck (Astrakhantsev et al., 2015), a majority of research focuses mainly on dominant European languages (e.g., English), leaving a gap in other lesser-resourced ones (e.g., Slovenian). To fill this gap, in our research, we verify the applicability of our approaches to not only the most dominant language in NLP research (e.g., English) but also other less-represented languages (e.g., French, Dutch, and Slovenian) (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022; H. T. H. Tran, Martinc, Pelicon, et al., 2022) using transformer-based models as the token classifier with different annotation regimes. One of our

solutions (fine-tuned SloBERTA extractor) has been integrated as term extracting feature into the Slovenian Terminology Portal[21] whose docker version can be found at https://github.com/honghanhh/ate-docker.

3. *Introducing novel NOBI annotation scheme to leverage the ability to extract nested terms (H. T. H. Tran, Martinc, et al., 2024).* In the traditional BIO regime, B stands for the beginning word in the term, I stands for the word inside the term, and O stands for the word not part of the term. The terms from a gold standard list are first mapped to the tokens in the raw text and each word inside the text sequence is annotated with one of three labels. However, it is not optimized for nested terminology extraction. Thus, we propose NOBI (H. T. H. Tran, Martinc, et al., 2024), an annotation regime with two additional labels BN and IN, where N refers to nested single-word terms, which can be at the beginning (BN) or inside (IN) position of a longer term. The guideline of the NOBI regime on ACTER and RSDO5 dataset can be found at https://github.com/honghanhh/nobi_annotation_regime.

4. *Verifying the impact of transfer learning across languages and domains.* Being aware of the challenges of collecting data for a specific domain and the lack of well-annotated corpora for less-represented languages, we are among the first to evaluate terminology systems in cross-lingual and multilingual learning tran2022can,tran2024can against the monolingual one. In cross-lingual learning, we evaluate the ability of the model to apply knowledge learned in one or more languages to terminology extraction tasks in another, unseen language. In multilingual learning, we investigate whether adding more data from other languages to the training set in the target language improves the predictive performance of the model. Interestingly, we find that a token classifier fine-tuned in one (e.g., English) or more languages (e.g., English, French) can predict the terms in new, unknown languages (e.g., Dutch) with competitive performance. However, the cross-lingual performance was less successful in Slovenian, which can be explained partly by Slovenian being a morphologically rich Slavic language and partly by dataset creation (as other datasets were annotated within the same annotation campaign). The source code can be found at https://github.com/honghanhh/ate-2022.

5. *Verifying the need for domain specificity in the era of large-scale language models through the introduction of the LlamATE system (H. T. H. Tran, González-Gallardo, Doucet, & Pollak, 2024).* Most of the current neural works considered terminology extraction as a sequence or token classification task. We approach the task with a new perspective on the generative model. In practical use where both well-annotated data and computational resources are not adequate for fine-tuning, with the advent of LLMs, we are the first to propose *promptATE* (H. T. H. Tran, González-Gallardo, Delauney, et al., 2024), and *LlamaATE* (H. T. H. Tran, González-Gallardo, Doucet, & Pollak, 2024), both of which consider LLMs as a predictor to query the candidate term without additional fine-tuning steps. We observe that for both systems, using only a few examples as the demonstration for the predictor, the system outperforms the fine-tuned version with the same training examples and provides competitive results against the fully-supervised token classifiers. Furthermore, *LlamaATE*, our 4-step pipeline with instruction prompting and additional self-verification steps, does not need to be explicitly told the domain of the examples in the prompt (explicit vs. implicit), both for examples coming from the same domain and for cross-domain examples. The pipeline offers a promising approach for low-resource terminology

[21] https://terminoloski.slovenscina.eu/

extraction tasks, which is useful and practical due to the limited well-annotated data. While it might not be a replacement for fully-supervised models, it can enhance efficiency and accuracy by streamlining the pre-annotation process and accelerating manual annotation efforts. The source code can be found at https://github.com/honghanhh/terminology2024.

1.3.2 Main Publications

This doctoral work has led to the following publications (note that we only listed the publication related to terminology extraction):

1. **Journals**:

   - (H. T. H. Tran, Martinc, et al., 2024) **Hanh Thi Hong Tran**, Matej Martinc, Andraz Repar, Nikola Ljubesic, Antoine Doucet, Senja Pollak. *Can Cross-domain Term Extraction Benefit from Cross-lingual Transfer and Nested Term Labeling?*. Machine Learning, 113(7), 4285-4314. 2024.
   - (Accepted) (H. T. H. Tran, González-Gallardo, Doucet, & Pollak, 2024) **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Antoine Doucet, Senja Pollak. "*LlamATE: Automated Term Extraction Using Large-scale Generative Language Models*". Computational Terminology Special Issue – Terminology, 2024.

2. **Conferences**:

   - (Accepted) (H. T. H. Tran, González-Gallardo, Moreno, et al., 2024) **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Jose Moreno, Antoine Doucet, and Senja Pollak. "*Is Domain Important for Automatic Term Extraction in the Era of Large Language Models?*". Terminologie & Ontologie : Théories et applications (ToTh 2024), 2024.
   - (Accepted) (H. T. H. Tran, González-Gallardo, Delauney, et al., 2024) **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Julien Delaunay, Jose Moreno, Antoine Doucet, and Senja Pollak. "*Is Prompting What Term Extraction Needs?*". 27^{th} International Conference on Text, Speech, and Dialogue (TSD 2024), 2024.
   - (Accepted) (Sun et al., 2024) Wenjun Sun, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Mickaël Coustaty and Antoine Doucet. "*LIT: Label-Informed Transformers on Token-based Classification*". 28^{th} International Conference on Theory and Practice of Digital Libraries (TPDL 2024), 2024.
   - (Delaunay et al., 2024) Julien Delaunay, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Jose Moreno, Antoine Doucet, and Senja Pollak. "*CoastTerm: a Corpus for Multidisciplinary Term Extraction in Coastal Scientific Literature*". 27^{th} International Conference on Text, Speech and Dialogue (TSD 2024), 2024.
   - (H. T. H. Tran, Martinc, Pelicon, et al., 2022) **Hanh Thi Hong Tran**, Matej Martinc, Andraz Repar, Antoine Doucet, Senja Pollak. "*Ensembling Transformers for Cross-domain Automatic Term Extraction*". International Conference on Asian Digital Libraries (ICADL 2022). Cham: Springer International Publishing, 2022.

- (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022) **Hanh Thi Hong Tran**, Matej Martinc, Antoine Doucet, Senja Pollak. "*Can Cross-domain Term Extraction Benefit from the Cross-lingual Transfer?*. International Conference on Discovery Science (DS 2022). Cham: Springer Nature Switzerland, 2022.
- (H. Tran et al., 2022) **Hanh Thi Hong Tran**, Matej Martinc, Antoine Doucet, Senja Pollak. "*A Transformer-based Sequence-labeling Approach to the Slovenian Cross-domain Automatic Term Extraction*". Slovenian Conference on Language Technologies and Digital Humanities (JTDH 2022), 2022.
- **Hanh Thi Hong Tran**, Matej Martinc, Antoine Doucet, Senja Pollak. "*Contextual and global sequential labeling approaches to automatic terminology extraction.* 14^{th} Jožef Stefan International Postgraduate School Students' Conference. 2022. (p.50).

Besides, we have the following publications under evaluation:

- Matej Martinc, **Hanh Thi Hong Tran**, Boskho Koloski, Senja Pollak. "*MOSES: Mixture of Specialized Experts for Supervised Extraction*". Transactions of the Association for Computational Linguistics (TACL 2024), 2024.
- (H. T. H. Tran et al., 2023) **Hanh Thi Hong Tran**, Matej Martinc, Jaya Caporusso, Antoine Doucet, Senja Pollak. "*The Recent Advances in Automatic Term Extraction: A survey*". ACM Computing Surveys, 2023.

3. Models and Products

Our Slovenian model using SloBERTa (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022) as the backbone was integrated into the Slovenian terminological portal[22]. The docker version can be found at https://github.com/honghanhh/ate-docker.

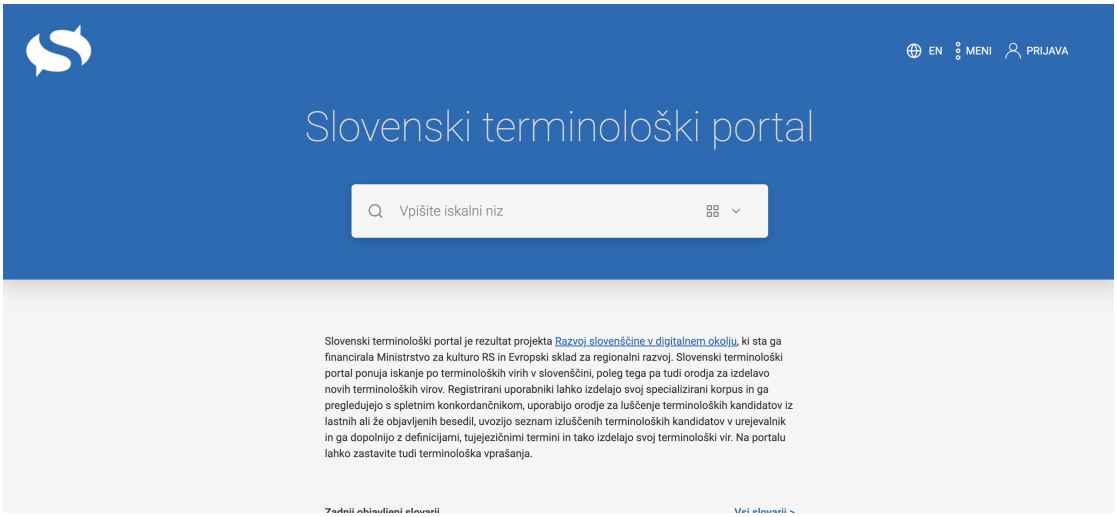


Figure 1.2: Slovenian terminological portal.

In addition, XLMR token classifiers that use NOBI for data annotation were published in the Terminology Extraction collection[23] on HuggingFace. The collection includes classifiers for monolingual, cross-lingual, and multilingual learning.

Other publications are listed in Chapter **??**.

[22]https://terminoloski.slovenscina.eu/

[23]https://huggingface.co/collections/tthhanh/terminology-extraction-ate-66a26e41d723c565bbb8922f

1.4 Thesis Structure

The remainder of this dissertation is organized as follows.

In Chapter 2, we present the related work on terminology extraction, including a systematic review of the available corpora, the evolution of DL-based approaches, and the evaluation metrics.

In Chapter 3, we demonstrate in detail two corpora, namely the *Annotated Corpora for Term Extraction Research* (ACTER)[24] and the *Corpus of term-annotated texts* (RSDO5)[25] datasets.

In Chapter 4, our focus is to provide proof to the general hypothesis that terminology extraction can benefit from Transformers models when formulated as a token classification or sequence labeling task (see H1). We demonstrate it in cross-domain settings in monolingual, cross-lingual, and multilingual learning scenarios, including in rich- and lesser-resourced languages. Moreover, we investigate the potential of adding additional layers on top of the base architecture to further improve the performance.

In Chapter 5, we highlight the obstacles of the standard annotation regimes in capturing the nested terms and introduce a novel nested term labeling mechanism for the ATE task, named NOBI (See H2). We compare the performance of the token classifier from Chapter 5 using the standard BIO regime and ours in both ACTER and RSDO5 datasets to demonstrate the impact of our mechanism.

In Chapter 6, we focus on generative models and test two different sets of methods to extract the candidate terms: [1] template-based ranking models and [2] large generative language models (LLMs) using prompt engineering with in-context learning. In the latter approach, we also verify the domain impact on the ATE in monolingual, cross-lingual, and multilingual learning.

For each hypothesis, in Chapters 4 to 6, we provide experimental evaluation and error analysis to better understand the behavior and characteristics of the classifier, as well as the obstacle that needs to be solved in future work.

In Chapter 7, we summarize our work with general findings and conclusions before listing all the papers published during my Ph.D. journey in Bibliography, including both publications that are directly and not directly related to this dissertation.

[24] https://github.com/AylaRT/ACTER

[25] https://www.clarin.si/repository/xmlui/handle/11356/1470

Chapter 2

Related Work

ATE has been a notoriously challenging task in the field of natural language processing (NLP). In recent years, several datasets and techniques have been constructed and developed for ATE tasks. Recent advances in supervised machine learning (ML) and deep learning (DL) approaches have prevailed over unsupervised approaches in the context of terminology extraction. As the traditional approaches have been comprehensively addressed in previous surveys (da Silva Conrado et al., 2014; Kageura & Umino, 1996), this section aims to provide a research trajectory in the field of supervised terminology extraction.

First, in Section 2.1, we provide a brief overview of the previous surveys and comparative studies on our specific tasks to get an overview of the existing work and to show what is missing in the previous studies.

Second, in Section 2.2, we give an overview of the resources for terminology extraction over the last decades. All well-annotated corpora are considered if the following two requirements are met: [1] they are publicly available, and [2] a description has been published.

Third, we propose a systematic review of ML- and DL-based approaches to terminology extraction and compare them with previous approaches based on feature engineering and shallow supervised learning algorithms in Section 2.3. Our comparative analysis highlights the improvements achieved by neural networks and shows that incorporating some insights from previous work on ATE systems based on feature engineering can lead to further improvements.

Fourth, in Section 2.4, we present all the metrics used in terminology extraction and categorize them according to either indirect (i.e., through a downstream application) or direct evaluation methodology (i.e., whether the evaluation was done by human judges, dictionary-based, or gold standard-based) and the scope of the results (i.e., whether we evaluate the entire results, parts of the results, or the best top-k results).

Fifth, in Section 2.5, we report the results for several ATE systems on the ACTER corpora by comparing the candidate term list extracted on the whole test set level with the manually annotated gold standard of each domain using a strictly matching F_1- score before jumping to the discussion about the current status of terminology extraction in Section 2.6.

2.1 Previous Surveys and Comparative Studies

The first comprehensive survey of approaches to ATE was written by Kageura and Umino (1996). It provided a systematic overview of the two main trends in the principles and methods of automatic term recognition and introduced a distinction between linguistic and statistical approaches. Pazienza et al. (2005) argued that all contemporary algorithms

employed linguistic methods as a filtering step. A decade later, da Silva Conrado et al. (2014) presented an overview of different approaches for ATE tasks, which can be divided into three sub-categories: statistical, linguistic, and hybrid (i.e., statistical and linguistic). They also did a brief review of different corpora in different domains used for terminology extraction, but only for Brazilian Portuguese. Regarding monolingual rich-resourced languages such as English, Astrakhantsev et al. (2015) presented a survey of existing notions of a term and its linguistic features. They formulated the definition of automatic terminology extraction, analyzed available approaches, and proposed a general ATE pipeline consisting of four consecutive steps: preprocessing, term candidate collection, term candidate scoring, and term candidate ranking.

In addition to these initial survey studies, a systematic comparison of various ATE systems with the same corpora was also carried out as part of several shared tasks and workshops. NTCIR (Kageura et al., 2000) is an evaluation-based project for information retrieval and terminology extraction released between 1998 and 1999. However, this study suffered from the limited number of participants and the absence of previous evaluation initiatives for computational terminology. Later, CoRReCT (Enguehard, 2003) introduced an interesting data set with two proposed systems to detect candidate terms (i.e., FASTR and SYRETE) and a new term recognition evaluation protocol called controlled indexing. A few years later, the Campagne d'Evaluation des Systèmes d'Acquisition des Ressources Terminologiques (CESART) (El Hadi et al., 2006), presented terminology extraction as one of three subtasks with a more user-oriented evaluation. This project proposed an interesting new protocol for terminology extraction that includes a gold standard list of terms and a corresponding domain-specific acquisition corpus. This setting is considered one of the better approaches to term annotation in recent terminology extraction studies.

The TermEval 2020 shared task on the extraction of monolingual automatic terminology, organized as part of the CompuTerm workshop (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020), presented one of the first opportunities to systematically study and compare various terminology extraction systems in comparable corpora in English, French and Dutch of four domains (*corruption*, *wind*, *equitation*, *heart failure*). The participating systems ranged from traditional approaches based on linguistic and statistical features to systems based on shallow machine learning and neural networks. While the workshop was an important step forward in systematic comparison among different methodologies, there is still room for improvement as the open source code for most of the participating systems is not available, which hinders their replicability.

2.2 Related Corpora

Table 2.1 covers manually annotated domain-specific resources with their latest version for terminology extraction systems developed over the past three decades. Note that we only report the data resources that meet four criteria: [1] they have been used for term evaluation; [2] they contain more than 100 terms; [3] they have annotations or gold standards; and [4] they are publicly available at a specific URL or easily to reconstruct based on the data description.

The monolingual datasets for terminology extraction task can be divided into two categories:

- Rich-resourced languages
  - GENIA (v3.0) (Kim et al., 2003): a collection of 2,000 English abstracts in the articles extracted from the MEDLINE database. 93,293 out of all 436,967 words were annotated as biological terms.

Table 2.1: List of most popular public open-sourced term datasets.

	Dataset	Year	Domain(s)	Language(s)	URL(s)
Monolingual datasets	GENIA	2003	Biomedicine	English	http://www.geniaproject.org/genia-corpus
	GO	2005	Gene	English	http://geneontology.org/
	CRAFT v2.0	2016	Biomedicine	English	https://bioNLP-corpora.sourceforge.net/CRAFT/
	ACL RD-TEC v1.0	2014	Computational linguistics	English	http://pars.ie/lr/acl-rd-tec-terminology
	ACL RD-TEC v2.0	2016	Computational linguistics	English	http://pars.ie/lr/acl_rd-tec
	JPED	2005	Pediatrics	Portuguese	https://repositorio.ufsc.br/handle/123456789/102257
	ECO	2005	Ecology	Portuguese	http://www.nilc.icmc.usp.br/nilc/projects/bloc-eco.htm
	Irish Wiki	2019	Education	Irish	https://github.com/jmccrae/irish_saffron
	Hindu Wiki	2022	Education	Hindu	https://hi.wikipedia.org/w/api.php
	RSDO5	2021	Ph.D. theses, Scientific book, Journal articles, Graduate textbooks	Slovene	https://www.clarin.si/repository/xmlui/handle/11356/1400
Multilingual datasets	TTC	2012	Wind energy, Mobile technology	Chinese English French German Latvian Russian Spanish	https://www.clarin.si/repository/xmlui/handle/11356/1463/
	BitterCorpus	2014	Information technology	English Italian	https://mt4cat.fbk.eu/benchmarks/bittercorpus
	TermFrame v1.0	2019	Karstology	Slovene Croatian English	http://termframe.ff.uni-lj.si/
	KAS-biterm	2019	Academic writing	Slovene English	https://www.clarin.si/repository/xmlui/handle/11356/1263
	ACTER v1.5	2020	Corruption, Dressage, Heart failure, Wind energy	English French Dutch	https://github.com/AylaRT/ACTER

  - Gene Ontology (GO) (Consortium, 2004): the structured controlled vocabularies of molecular and cellular biology that are freely available for community use in the annotation of genes, gene products, and sequences. Between 2005 and 2022, a total of 7,694,564 annotations were made, with 43,329 valid terms, 4.024 obsoleted terms, and 2,438 merged terms.
  - CRAFT (v2.0) (Bada et al., 2012): a collection of English full-text, open-access biomedical journal articles, each of which is a member of the PubMed Central Open Access Subset[1]. The dataset includes around 100,000 ontologies/terms annotated from 67 out of the 97 articles in seven different categories.
  - ACL RD-TEC (QasemiZadeh & Handschuh, 2014): an English manually annotated terms from ACL Anthology Reference Corpus for evaluating the terminology extraction and classification from literature in the domain of computational linguistics (v1.0: 82,000 terms, v2.0: 300 unique abstracts with classified terms).
  - JPED (Coulthard et al., 2005): a Brazilian Portuguese collection of 283 texts from the Pediatrics Journal (Jornal de Pediatria) and a gold standard of 1,534 bigrams and 2,647 trigrams.
  - ECO (Zavaglia et al., 2005): 390 Portuguese documents about the domain of Ecology in the context of the BLOC-Eco project.

- Lesser-resourced languages
  - RSDO5[2]: a collection of 12 Slovene texts collected between 2000 and 2019, from

[1] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/
[2] https://www.clarin.si/repository/xmlui/handle/11356/1400

the fields of biomechanics, linguistics, chemistry, and veterinary science. Over 250,000 words and almost 38,000 manually annotated terms were included.
  - Irish Wiki (McCrae & Doyle, 2019): a collection of 11 Wikipedia Irish documents of 5,178 words and 864 extracted terms
  - Hindi Wiki (Banerjee et al., 2022): a collection of 71 Wikipedia Hindu documents comprising 11,960 words and 953 manually annotated terms.

The monolingual datasets differ considerably, as there is a lot of variation in both the annotation and evaluation. Diverse approaches have been applied to each different aspect, including annotation type (candidate term list or just source text, etc.), annotation scheme (binary or multi-label, etc.), annotation guidelines (term length, PoS patterns, whether or not to add named entities, etc.). Another problem is that each monolingual dataset covers a different domain and usually covers only a limited number of terms within that specific domain. Consequently, it is extremely difficult to make any comprehensive comparison among corpora or combine multiple datasets into an extensive corpus.

With the development of multilingual tasks (Devlin et al., 2018; Lang et al., 2021), datasets that cover multiple languages and domains are becoming increasingly important for the training and evaluation of multilingual and cross-lingual models.

- TTC (Daille, 2012; Gornostay et al., 2012): one of the first manually annotated texts from two domains (Wind energy and Mobile technology) in seven languages (Chinese, English, French, German, Latvian, Russian, and Spanish).

- BitterCorpus (Arcan et al., 2014): a manually annotated collection of parallel English-Italian documents with 874 domain-specific bilingual terms in the Information Technology (IT) domain extracted from the GNOME and KDE collections.

- TermFrame (v1.0) (Pollak et al., 2019; Š. Vintar et al., 2019; Vrtovec et al., 2019): a specialized corpus of karstology literature in Slovene, Croatian, and English. The terms were extracted from 24 English and 60 Slovene documents by comparing the domain corpus to the reference corpus, and with additional processing steps (e.g., filtering based on nested terms, stopwords, and fuzzy matching).

- KAS-term (Ljubešić et al., 2019): a collection of 700 bilingual Ph.D. theses (40 million tokens) in the KAS corpus about Slovene academic writing of three domains: Chemistry, Computer Science, and Political Science. The terms were labeled using three labels (terms, partial terms, abbreviations) in three categories of languages (Slovene, English, or other languages).

- ACTER (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020): the first meticulously annotated corpus that covered multiple languages and domains and that offered available up-to-date documentation and transparent annotation guidelines. Texts from four different domains (*corruption*, *dressage*, *heart failure*, and *wind energy*) and three languages (English, French, and Dutch) were manually annotated. ACTER solves the debate about whether or not to consider named entities as terms by providing two different versions of manual annotations: one containing only terms, and the other containing both terms and named entities.

In our investigation of publicly available corpora, we figured out that most original studies provide detailed descriptions of the initial versions of their datasets. While some papers incorporated both data and method descriptions, they often neglected to specify the data versions employed or whether the entire dataset was utilized or only a part. This inconsistency makes it difficult to thoroughly compare methods using the same corpora, especially when the latest version undergoes significant changes from the previous one.

2.3 Term Extraction Approaches

In this section, we offer an extensive overview of recent ATE systems, which tend to rely either on shallow ML techniques or deep neural networks. In most cases, traditional approaches to terminology extraction do not rely only on ML. Instead, they tend to extract candidate terms above a certain threshold, derived from several linguistic and statistical features. Traditional approaches are not covered in our evolution in methodologies, since they have been covered in previous surveys (Astrakhantsev et al., 2015; Kageura & Umino, 1996). In this section, starting from the advent of ML up until now, we present ML-based (neural and non-neural) ATE systems, that have not been systematically described and compared in any recent survey study.

2.3.1 Machine Learning Approaches

Despite some variations in features and models, most supervised ML approaches for ATE follow the traditional approach, which includes three main steps: [1] preprocessing, [2] feature engineering, and [3] term extraction model as a classifier. In the preprocessing step, some operations are performed on the input texts (e.g., sentence segmentation, word segmentation, and PoS tagging) to prepare the input text for further steps. In the feature engineering step, we describe the candidate terms by different features (see examples of different types of features in Figure 2.1). In the final steps, the features are fed into the ML classifiers so that the classifiers can learn from the training set and use this knowledge to make predictions for new, unseen text.

Due to their relatively low accuracy, these first ML approaches were used mainly to complement approaches based on hand-crafted rules. This, along with the success of traditional systems (e.g., TermoStat (Drouin, 2003)), which relied on several linguistic and statistical features, led to the idea of combining different types of information. That is, multiple linguistic and statistical termhood indicators were fed as features to a variety of ML algorithms. While several NLP tools (e.g., tokenization, lemmatization, stemming, chunking, and PoS tagging) are employed in this approach to obtain linguistic profiles of term candidates, numerous statistical measures are also applied to this approach, including termhood (S. Vintar, 2010), unithood (Daille et al., 1994), C-value (K. T. Frantzi et al., 1998). Regarding ML algorithms, the most popular algorithms used for ATE include AdaBoost (Castellví et al., 2001), ROGER evolutionary algorithm (Azé et al., 2005), RIPPER rule induction (Foo & Merkel, 2010), CRF++ (Judea et al., 2014), K-nearest neighbors (Qasemizadeh & Handschuh, 2014), Logistic Regression (Bolshakova et al., 2013; Dobrov & Loukachevitch, 2011; Fedorenko et al., 2014; Loukachevitch, 2012; Nokel et al., 2012), Decision Trees (DTs) (Karan et al., 2012), and Support Vector Machines (SVM) (Ljubešić et al., 2018).

An example of an ML approach employing extensive feature engineering and several classifiers is given in Conrado et al. (2013) where the authors proposed to select statistical and linguistic features and feed them into different ML classifiers (e.g., JRip, Naive Bayes, J48, or SMO from WEKA). Y. Yuan et al. (2017) instead used common features (e.g., term frequency, c-value, weirdness) extracted from the token n-grams (n=1,2,3,4,5) excluding all the stopwords and fed them into different classifiers: Random Forest (RF), Linear SVM, Multinomial Naive Bayes, Logistic Regression, and SGD classifiers.

With the advent of the newly annotated ACTER corpora, one of the TALN-LS2N approaches (Hazem et al., 2020) used the combination of different meaningful information—such as linguistic, stylistic, statistic, and distributional descriptors—to generate the feature vectors. Then it used the XGBoost model to learn several classifiers, which were weighted according to their performance and iteratively aggregated. HAMLET (Rigouts

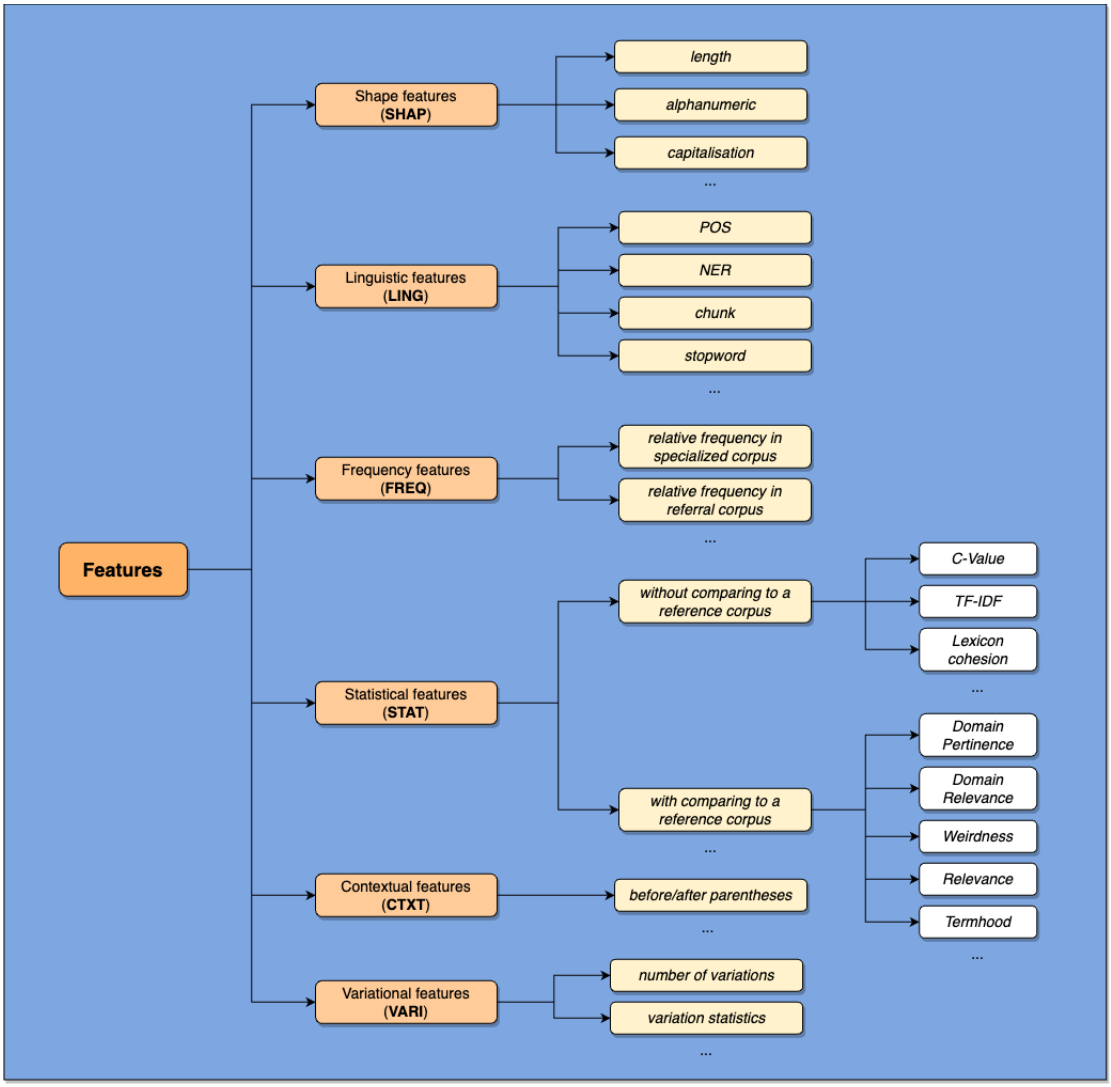


Figure 2.1: Feature groups and subgroups for ML models (Rigouts Terryn et al., 2021).

Terryn et al., 2021) included a relatively wide range of supervised ML methods (e.g., DTs, RF, Multi-layer Perceptron, and Logistic Regression) and relied on a total of 152 features from six different feature groups: morphological, frequency-based, statistical, relational and linguistic, and corpus-based. The Binary RF classifier performed the best of all tested classifiers. Furthermore, Nugumanova et al. (2022) combined probabilistic topic modeling (PTM) and non-negative matrix factorization (NMF). They compared five different NMF algorithms and four different NMF initializations and found optimal combinations of NMF to compare with the extraction baseline (e.g., TF-IDF, RAKE, YAKE, and TextRank).

2.3.2 Deep Learning or Neural Approaches

Using neural networks, especially language models, to solve ATE tasks, has gained more traction in recent years. Their application is performed to represent the information in the text with word embeddings or to apply a deep architecture as an end-to-end classifier.

2.3.2.1 Neural-based Embeddings

The embeddings for terminology extraction are often pre-trained on a large general corpus, and then potentially fine-tuned on specific corpora during classification.

*Static Embeddings.*The most popular general static or non-contextual word embeddings are GloVe embeddings (Amjadian et al., 2016, 2018; Kucza et al., 2018; N. T. Le & Sadat, 2021; Zhang, Petrak, & Maynard, 2018), followed by the domain-specific Word2Vec embeddings (Zhang, Gao, & Ciravegna, 2018), either employing the CBOW or the skip-gram architecture (Amjadian et al., 2018; Bay et al., 2021; R. Wang et al., 2016), and FastText (Terryn et al., 2022). Meanwhile, some studies explored the idea of concatenating general and domain-specific embeddings (Amjadian et al., 2018; Bay et al., 2021; Hätty et al., 2020). These embedding concatenation techniques demonstrated that any strategy using both types of embeddings performed better than those only using either the general or the domain-specific ones but failed to capture contextual information.

Contextual Embeddings. Due to the significance of general knowledge and contextual information encoded in pre-trained language models for the downstream tasks, several contextual embeddings have been proposed for the ATE task (Andrius, 2020; Terryn et al., 2022). These include Flair embeddings[3], which are neural, character-based language models from the FlairNLP framework developed for multiple languages and incorporating subword information. Furthermore, transformer-based embeddings have been explored, some of which have already been tested in terminology extraction, for example, BERT embeddings, and stacked Flair + BERT embeddings (Andrius, 2020; Terryn et al., 2022).

2.3.2.2 Neural-based Architectures

The use of deep neural networks in ATE is not only limited to the generation of embedding representations. Neural architectures are also used as end-to-end terminology extraction systems, which, depending on the nature of the neural methods applied to the ATE task, are divided into three main types, as demonstrated in Figure 2.2: [1] (binary) sequence classifiers; [2] token classifiers; and [3] sequence-to-sequence (Seq2Seq) generation models.

Binary Sequence Classifiers. When deep neural models were first adopted, terminology extraction was considered a binary classification task, which assigns a binary label $y \in Y = \{is\_a\_term, not\_a\_term\}$ to each sequence s in a given sentence $S = \{s_1, ..., s_n\}$, where n denotes the number of sequences generated from all possible n-grams of a fixed length of a given sentence. Different BERT-based variants have been tested (e.g., RoBERTa for English, CamemBERT for French, and XLMR (Hazem et al., 2020; Lang et al., 2021)). While this method demonstrated superior performance than other ML-based approaches, generating all possible n-grams from every sentence across all documents for training purposes poses computational and storage challenges. As a result, subsequent studies embraced token classification as an alternative strategy.

Token Classification. A common approach is to consider ATE as a token classification task, which assigns a label $y \in Y = \{B, I, O\}$ to each word x in a given sentence $X = \{x_1, ..., x_n\}$, where Y denotes the set of labels in BIO regimes (see the explanation in Chapter 5), and n denotes the length of the given sentence. Several language models have been applied as token classifiers for ATE tasks, including RNN (Kucza et al., 2018), LSTM-CNN (R. Wang et al., 2016), CNN-BiLSTM-CRF (Han et al., 2018), using different embeddings as input to feed into LSTM (Andrius, 2020), BiLSTM (Andrius, 2020; N. T. Le & Sadat, 2021), LSTM-CRF (Andrius, 2020), and BiLSTM-CRF (Andrius, 2020; Hazem et al., 2022; Rigouts Terryn et al., 2022), respectively. However, with the advent of transformers, before our studies, only XLMR was initially proposed as a token classifier

[3]https://github.com/flairNLP/flair

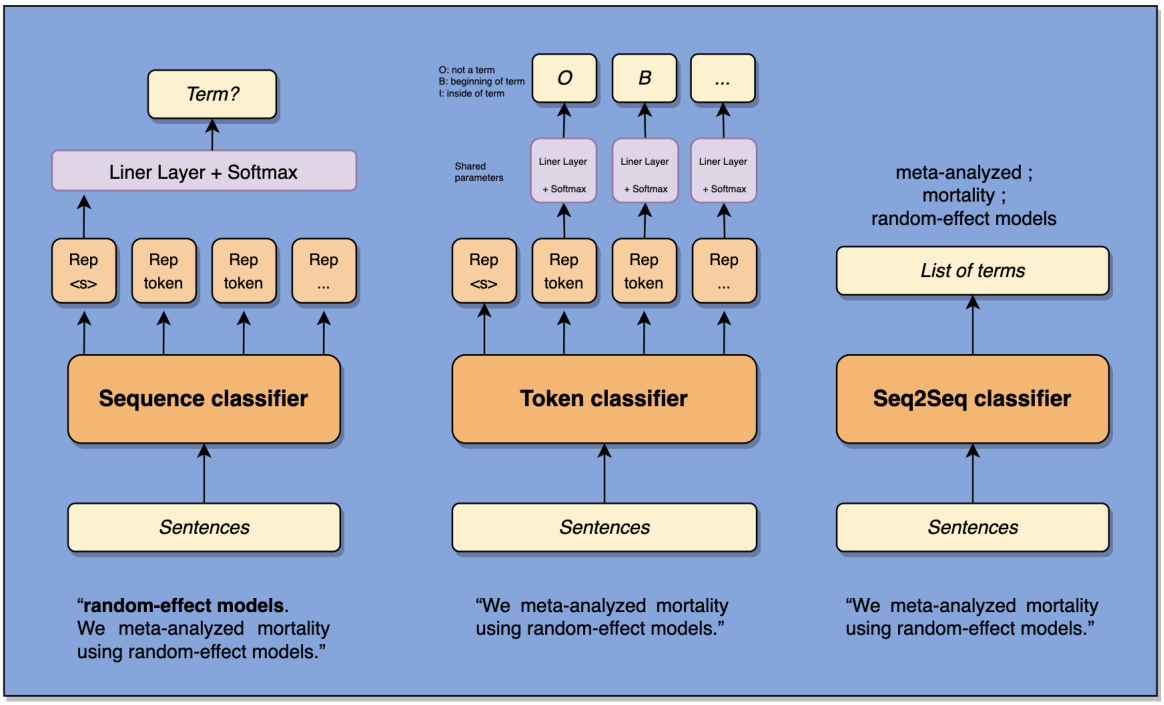


Figure 2.2: An example of how three types of neural classifiers work (from left to right: Sequence classifier; Token classifier; Seq2Seq classifier).

(Hazem et al., 2022; Lang et al., 2021) for English, French, and Dutch. Since this approach operates without upfront n-gram generation and handles each sentence as a single example that needs to be labeled, it is considerably more time-efficient than the previous approach.

Seq2Seq Classifiers. Despite the widespread use of Seq2Seq generation models for other downstream NLP tasks, the adoption of self-supervised pre-training approaches for ATE has just recently begun to gain traction. Lang et al. (2021) was the first to employ the sequence generation model mBART (Y. Liu et al., 2020) to transform the input sentences to sequences of comma-separated terms. This approach was inspired by the neural machine translation-based ontology learning proposed by Petrucci et al. (2018). While promising, the performance and applicability of this approach require additional testing and remain an unsolved issue.

2.4 Evaluation Metrics

ATE models usually provide a list of candidate terms from the given domain-specific corpus as the final output, and it is important to define the correct methods and metrics to evaluate the quality of this output. There are several variations among the evaluation methods, including both intrinsic and extrinsic mechanisms. The extrinsic methods assess the quality of the extracting system by measuring the improvement in the performance of another system or application that uses the results of terminology extraction as described in Vivaldi and Rodríguez (2007). Meanwhile, the intrinsic ones measure the quality of terminology extractors by evaluating some intrinsic properties, which are independent of their intended planned use. We can classify the direct evaluation approaches into two main aspects: the evaluation methodology and the scope of the results (Zhang et al., 2008) as shown in Figure 2.3. The evaluation methodology focuses on whether the evaluation

is performed by human judges, or whether it is dictionary-based or gold standard-based. Meanwhile, the scope of results answers the question of whether we evaluate the entire results, parts of the results, or the top-k results. Due to this variety in evaluation types, the performance of different approaches is often not directly comparable.

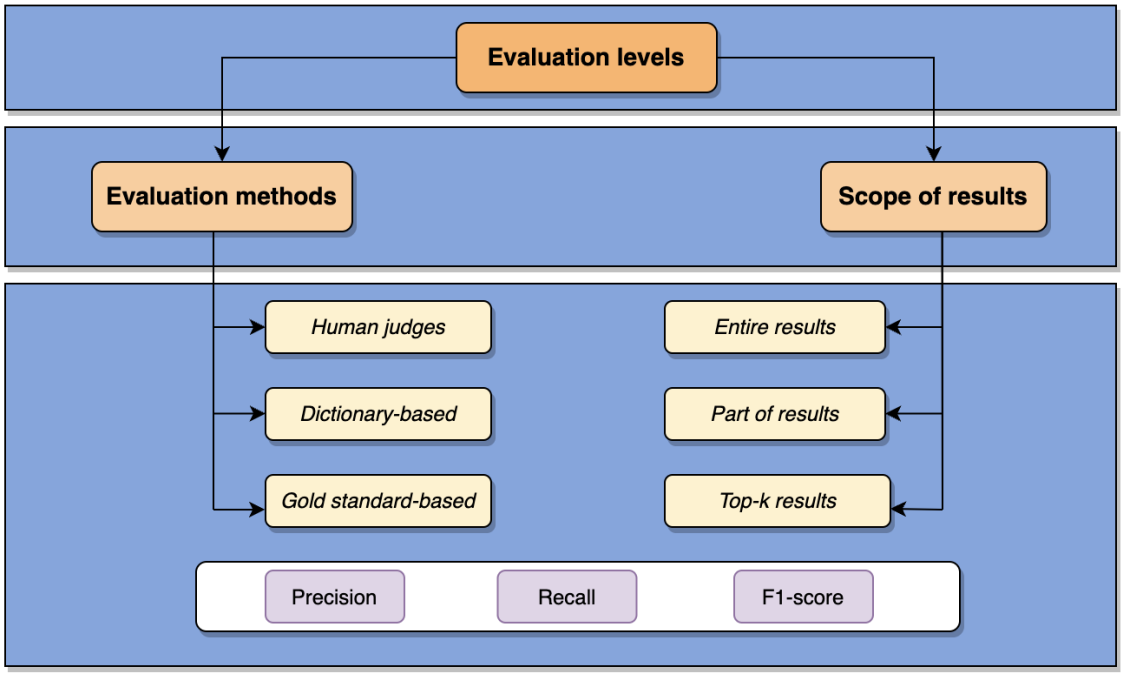


Figure 2.3: Overview of different evaluation metrics in ATE task.

Regarding the evaluation methods, in the initial studies about ATE tasks, having a human judge was one of the first approaches to evaluate how well the candidate terms were extracted. Justeson and Katz (1995) evaluated their system by asking a domain expert (i.e., a terminologist) to judge whether the extracted candidates were domain-specific terms.

To reduce the human effort, the dictionary-based evaluation was proposed either as a list or a dictionary of the pre-defined criteria to map with the extracted terms. Kageura and Umino (1996) used statistical features, e.g., termhood and unithood, for evaluation. L'Homme et al. (1996) applied five pre-evaluation criteria from the basic design evaluation to present how the results are shown in the candidate lists. Macken et al. (2013) identified the terms with linguistic preprocessing steps and matched the candidate terms to a pre-defined dictionary of PoS patterns.

Another popular approach is based on reference term lists, or the so-called gold standard corpora, which can be an adaptation of a pre-existing list, a sample (seed terms), or the list of all the terms in the corpus. The pre-existing term list (Dobrov & Loukachevitch, 2011; Wermter & Hahn, 2005) is not collected directly from the training corpora but is an already existing and community-wide terminology. Thus, the terminology extraction approaches using the corpora with this gold standard often evaluate how many candidate terms were actual terms but fail to measure how many terms in the text were correctly extracted. Meanwhile, the approach of considering a sample as the gold standard (Baroni & Bernardini, 2004; Loginova et al., 2012) took advantage of a web crawler to collect the domain-specific texts (e.g., TTC project). The crawler took a list of domain-specific words, the so-called seed terms, as input, and as outputs, the texts found on the Web in the domain of interest. The seed terms are usually term representative of the domain for

which we want to retrieve the web documents. To resolve the existing issues, the list of all the terms in the corpus (Kim et al., 2003; Rigouts Terryn, Hoste, Drouin, & Lefever, 2020) was used as a gold standard where the list was annotated directly from the corpora. This became a benchmark for manual annotation in terminology extraction until now. If this gold standard corpus is used, the final evaluation score is usually obtained by calculating precision (i.e., how many candidate terms were actual terms), recall (i.e., how many terms in the text were correctly extracted), and F_1-score or F_1 for short (i.e., the harmonic mean of precision and recall) for the obtained list of term candidates. We will specify these metrics when discussing the evaluation of different scopes of results.

Finally, in addition to the performance of the ATE system with respect to the evaluation metrics described above, some claim that other factors, such as the consistency of the candidate terms predicted, are also important. For example, Sauron (2002) proposed that the measures of a model's quality should also concern reliability, efficiency, maintainability, usability, and portability.

Regarding the scope of results, the most common ATE evaluation approach is to compare the candidate term lists obtained by the system against the list of terms extracted by human annotators (i.e., the so-called gold standard corpus (Conneau et al., 2019; Kim et al., 2003; Rigouts Terryn, Hoste, Drouin, & Lefever, 2020; Rigouts Terryn et al., 2021)). This is done by calculating precision (see equation 2.1), recall (see equation 2.2), and F_1-score (see equation 2.3).

$$Precision = \frac{Number\ of\ correctly\ extracted\ terms}{All\ the\ candidate\ terms} = \frac{True\ Positives}{True\ Positives + False\ Positive} \tag{2.1}$$

$$Recall = \frac{Number\ of\ correctly\ extracted\ terms}{All\ terms\ in\ the\ corpus} = \frac{True\ Positives}{True\ Positives + False\ Negative} \tag{2.2}$$

$$F_1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.3}$$

Several ATE systems use the same evaluation approach as the keyword extraction task, i.e., they choose the top-k best candidate terms for evaluation (top-100 candidate terms (Ideue et al., 2011; Kupiec, 1993), top-300 candidate terms (S. Vintar, 2010), top-500 candidate terms (Daille, 1994)). Macken et al. (2013) and (Zhang, Petrak, & Maynard, 2018) evaluated the best-k candidate terms by using a variable k. Most of these studies employed Precision for evaluation of the best-k terms (Drouin, 2003; Haque et al., 2018; Macken et al., 2013; Repar et al., 2019a; Sclano & Velardi, 2007). One notable insight that emerged from our literature review is that the number of studies using just Precision is higher than the number of studies that evaluate the system according to all three evaluation criteria (i.e., precision, recall, and F_1-score). This is due to the relative lack of gold standard data available in the past. However, thanks to the more recent effort of constructing and manually annotating domain-specific datasets for ATE, most of the current systems employed on contemporary benchmark datasets (e.g., ACTER) are evaluated according to all three metrics (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020; Rigouts Terryn et al., 2021).

2.5 Comparative Evaluation

Since terminology extraction methods vary greatly concerning definition, corpora, domains, languages, and evaluation metrics, a comparative evaluation of terminology extraction methods is hardly achievable. Nevertheless, in this section, we present the results for several ATE systems on the ACTER corpora by comparing the candidate term list extracted on

Table 2.2: F_1-score evaluation of benchmark approaches on the *heart failure* test set from the ACTER corpus in three languages (English (EN), French (FR), and Dutch (NL)) and two types of annotation, with named entities (NES) and without named entities (ANN). The best approach for each category is highlighted in bold.

Methods	**ACTER**					
	ANN			**NES**		
	EN	FR	NL	EN	FR	NL
Non-neural models						
RACAI (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020)	39.3	-	-	41.3	-	-
e-Terminology (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020)	21.4	20.6	15.3	20.1	19.7	14.4
NYU (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020)	31.5	-	-	30.6	-	-
HAMLET (Rigouts Terryn et al., 2021)	**54.2**	**60.2**	**66.1**	**55.4**	**60.8**	**66.0**
Feature-based XGBoost (Hazem et al., 2022)	-	-	-	33.6	50.9	34.1
NMF (Nugumanova et al., 2022)	33.5	30.9	30.1	33.7	30.7	30.3
Monolingual deep learning models						
TALN-LS2N (Hazem et al., 2020)	45.0	45.9	-	46.7	48.1	-
NLPLab UQAM (N. T. Le & Sadat, 2021)	17.8	12.9	18.6	18.1	13.2	18.7
XLMR Sequence Classifier (Lang et al., 2021)	-	-	-	45.2	46.0	48.5
XLMR Token Classifier (Lang et al., 2021)	-	-	-	58.3	52.9	**69.6**
mBART NMT (Lang et al., 2021)	-	-	-	53.2	55.9	65.2
Vanilla-biLSTM-CRF (Hazem et al., 2022)	-	-	-	8.17	6.53	7.50
BERT (Hazem et al., 2022)	-	-	-	48.2	-	-
CamemBERT (Hazem et al., 2022)	-	-	-	-	51.1	-
BERT (NER) (Hazem et al., 2022)	-	-	-	37.4	-	51.0
CamemBERT (NER) (Hazem et al., 2022)	-	-	-	-	51.1	-
BERT-biLSTM-CRF (Hazem et al., 2022)	-	-	-	29.5	25.6	27.4
mBERT (Hazem et al., 2022)	-	-	-	45.8	47.2	-
Multilingual deep learning models						
XLMR Sequence Classifier (Lang et al., 2021)	-	-	-	46.0	46.7	56.0
XLMR Token Classifier (Lang et al., 2021)	-	-	-	**56.2**	55.3	**67.8**
mBART NMT (Lang et al., 2021)	-	-	-	55.3	**57.6**	64.9
mBERT (Hazem et al., 2022)	-	-	-	45.4	44.9	51.0
Cross-lingual deep learning models						
XLMR Sequence Classifier (Lang et al., 2021)	-	-	-	44.7	48.1	58.0
XLMR Token Classifier (Lang et al., 2021)	-	-	-	**58.3**	57.6	**69.8**
mBART NMT (Lang et al., 2021)	-	-	-	55.2	57.4	59.6

the whole test set level with the manually annotated gold standard of each domain using a strictly matching F_1-score. We chose to report the performance on this dataset because it is the most systematically annotated corpus that covers multiple languages and domains, and that also contains available up-to-date documentation and transparent annotation guidelines (Rigouts Terryn, 2021) with a high inter-annotator agreement (IAA) score (see the details on Chapter 3). The results of the evaluation are presented in Table 2.2.

TALN-LS2N, RACAI, NYU, e-Terminology, and NLPLab_UQAM were all competitors in the TermEval 2020 shared task (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020). The TALN-LS2N approach is the winning solution on the English and French datasets, whereas NLPLab_UQAM is the winning team on the Dutch corpus. While RACAI, NYU, and e-Terminology rely on feature engineering and statistical approaches, TALN-LS2N and NLPLab UQAM applied neural network-based models. After the competition, several new approaches were developed. For example, HAMLET is a novel supervised method inspired by traditional hybrid systems such as TermoStat. Furthermore, Nugumanova et al. (2022) combined probabilistic topic modeling (PTM) and non-negative matrix factoriza-

tion (NMF), and the optimal combinations of NMF outperformed four baseline extraction methods (e.g., RAKE (Rose et al., 2010), YAKE (Campos et al., 2020), and TextRank (Mihalcea & Tarau, 2004)). All the described approaches were implemented on the ACTER dataset of version 1.2, which was first released by the TermEval 2020 shared task. However, none of them had considered the problem as a token classification task and had taken advantage of the transformer-based models. Later, Lang et al. (2021) proposed a comprehensive comparison between three transformer-based ATE models operating at the sentence level: token classification (or sequence-labeling) task, sequence classification, and sequence-to-sequence (Seq2Seq) generation. Their approaches were applied to the ACTER dataset of version 1.5. However, there were not many changes to actual annotations, but major updates to how the annotations were presented.

Most deep learning-based approaches also proved to be very competitive and surpassed the performance of most non-neural methods by a large margin on all languages and annotation types. Lang et al. (2021) also showcased the potential of cross-lingual learning when considering ATE tasks as a token classification problem. These approaches currently represent the new state-of-the-art (SOTA) methods regarding most languages in the ACTER corpus.

2.6 Discussion

We summarized the recent advances in the automatic terminology extraction task, covering both classic ML models based on feature engineering and novel neural network models that have yielded several important insights, especially with the advent of pre-trained language models. We first surveyed different resources and presented a systematic list of well-annotated monolingual and multilingual corpora for the ATE task. Then, we presented the first systematic summary of DL-based approaches and compared their performance with ML-based approaches. Furthermore, we indicate all metrics used in the ATE task and categorize them according to the evaluation methodology and the scope of the results.

The main findings were that neural models generally outperformed ML models based on feature engineering by a large margin and established SOTA methods. However, there is still a gap between current neural approaches for ATE tasks and the development of natural language processing algorithms, especially for non-English corpora and there is room for improvement in performance. The winning solutions (Hazem et al., 2020) and the latest research (Lang et al., 2021) mostly considered the ATE tasks as binary sequence classification tasks using neural models. Therefore, they suffered from computational and storage challenges to generate all possible n-grams from each sentence as the input. Meanwhile, the majority of works considered the task as a token classification task but used classical models (e.g., LSTM, BiLSTM, LSTM-CRF, and BiLSTM-CRF (Han et al., 2018; Hazem et al., 2022; N. T. Le & Sadat, 2021)), leaving the gap in exploiting the emergence of transformers and the latter (large) language models. Furthermore, current research has not yet fully resolved the challenges discussed in Chapter 1, so we have the potential to further develop our methods, especially for ATE tasks in cross-domain settings where the application of the results to products is still in question. In the next chapter, we take a closer look at the corpora we use to develop our methods and test our hypotheses.

Chapter 3

Datasets

This chapter introduces two corpora that were used for the experimental validation of this doctoral work. *The Annotated Corpora for Term Extraction Research (ACTER)* and the *Slovenian Corpus of Term-annotated Texts (RSDO5)*. Both datasets met our standard requirements, including [1] they were used for terminology extraction; [2] they have annotations or gold standards; [3] they cover at least four different domains for cross-domain terminology learning; and [4] they are publicly available with well-documented annotation guidelines. Moreover, the ACTER corpora include three different Indo-European languages (e.g., English, French, and Dutch), which support us in verifying our hypothesis for monolingual, cross-lingual, and multilingual learning. RSDO5, on the other hand, focuses on Slovenian and helps us to verify our method in the lesser-known Slavic language.

3.1 The Annotated Corpora for Term Extraction Research (ACTER)

3.1.1 Description

The ACTER corpora, which stands for *Annotated Corpora for Term Extraction Research* (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020), is a collection of domain-specific corpora in which terms have been manually annotated. The dataset contains comparable trilingual corpora (English, French, and Dutch) in all four different domains: *corruption* (corp), *equitation* (equi), *heart failure* (htfl), and *wind energy* (wind). The data in the same domain are similar in subject, style, and length for each language, but they are not translations. The dataset has two types of gold standards: one that contains both terms and named entities (NES); and the other that contains only terms (ANN).

Figure 3.1 illustrates an example of the key difference between the ACTER's ANN and NES versions of gold standards. Given the sentence "*... This study uses the Medicare Patient Safety Monitoring System ...*", the gold standard of the ANN version consists of only the term "*Patient*" as the only term that was annotated as the ground truth. However, the gold standard of the NES version includes the Named Entity (NE) "*Medicare Patient Safety Monitoring System*" since both domain-specific terms and NEs were annotated in the ground truth.

Only a small part of the corpora was annotated by several annotators in order to calculate the inter-annotator agreement (IAA). Specifically, the IAA was calculated between three annotators who were not domain experts but were fluent in three languages. Each of them annotated about 3,000 tokens per language in the corpora over *corruption*, *heart failure*, and *wind energy* ($\pm$ 40,000 tokens in total). There were two rounds of annotation where the match was calculated based on the type and not the token. While the earlier

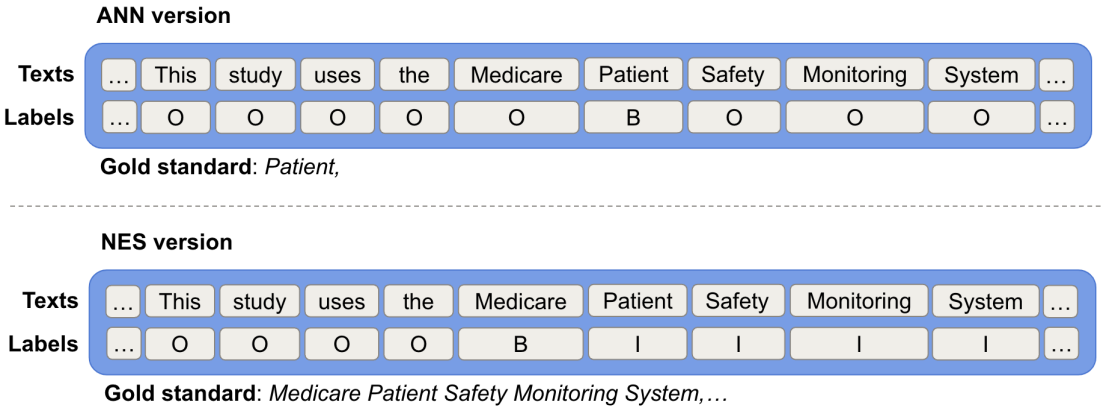


Figure 3.1: An example of ACTER's ANN and NES versions were annotated in the BIO regime.

iteration yielded an average F_1 of 0.614 and a Cohen's Kappa of 0.749, the later iteration yielded an average F_1 of 0.895 with a Cohen's Kappa of 0.927 after improving the guide to determine agreement on labels. Although language students helped with the annotation, most of the annotation work was done by a single annotator (Ayla Rigouts Terryn). In addition, all annotations by other annotators were reviewed by this main annotator (a terminologist fluent in all three languages). Only the terms on which several annotators agreed were retained.

Originally, the gold standards provided four labels, namely specific term, common term, out-of-domain (OOD) term, and additional named entities for the NES version. Since the TermEval shared task (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020) was set up as a binary task, all the term labels were combined and considered true terms. All participating and later researchers proposed the system based only on these two binary interpretations, and the four labels were made available afterward for a more detailed evaluation of the results. The gold standard lists of terms were ordered alphabetically, with no relation to their labels or degree of termhood. The details of the dataset have already been described in detail in Terryn et al. (2020) and Rigouts Terryn, Hoste, Drouin, and Lefever (2020), and we refer interested readers to this work for further information.

Table 3.1: ACTER corpus counts (only annotated parts of corpus).

Domain	Language	# files	# sentences	# tokens (incl. EOS)	# ANN terms	# NES terms
corp	en	12	2,002	54,849	927	1,172
	fr	12	1,977	63,084	979	1,207
	nl	12	1,988	56,221	1,047	1,287
equi	en	34	3,090	64,383	1,155	1,561
	fr	78	2,809	66,679	961	1,176
	nl	65	3,669	63,788	1,393	1,541
htfl	en	190	2,432	60,331	2,361	2,556
	fr	210	2,177	59,381	2,228	2,357
	nl	174	2,880	60,726	2,074	2,215
wind	en	5	6,638	71,042	1,091	1,529
	fr	2	4,770	74,529	773	967
	nl	8	3,356	62,040	940	1,229

3.1.2 Data Structure

The up-to-date ACTER dataset (version 1.5) has been structured as follows:

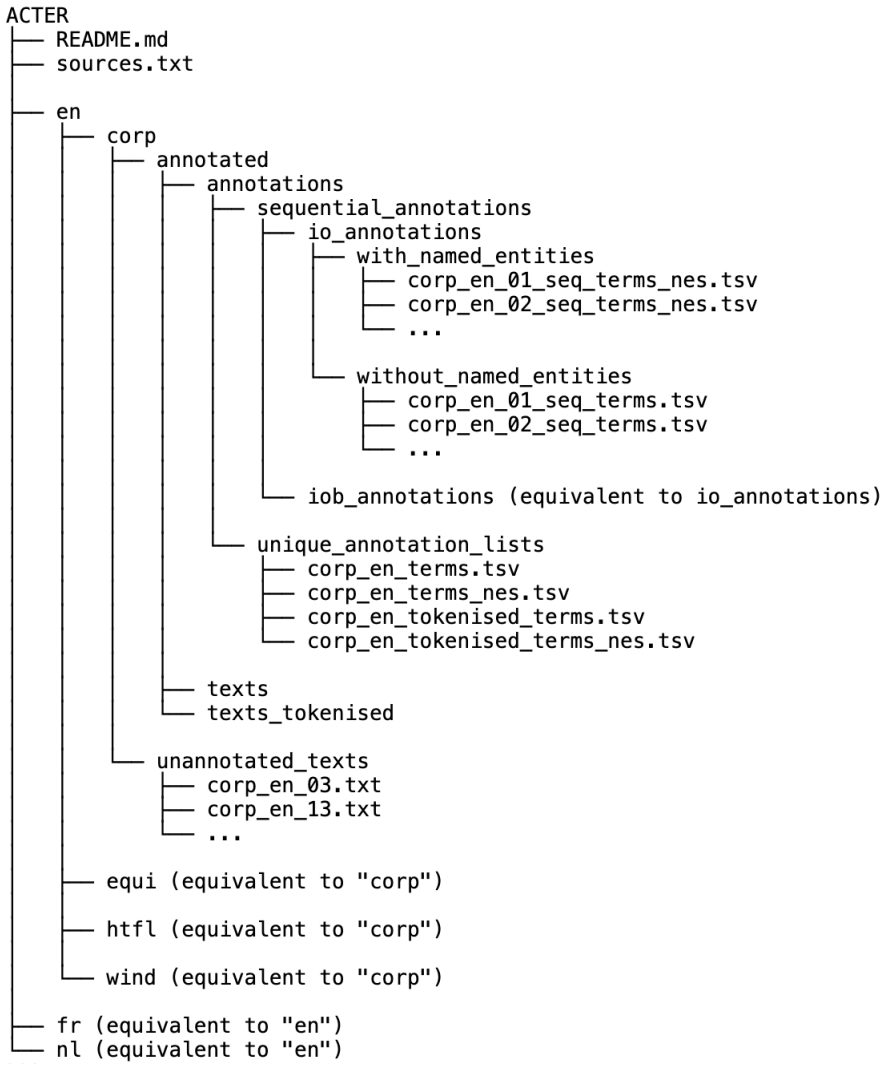


Figure 3.2: The overview of the ACTER corpus repository.

- **README.md, sources.txt**: These files provide information about the data sources.
- **languages** and language/**domains**: At the first level, there is one directory per language with an identical structure of sub-directories and files per language. At the second level, there are four directories, i.e., one per domain, each with an identical structure of sub-directories and files.
- language/domain/**unannotated_texts**: Per domain, there are annotated and unannotated texts. For the unannotated texts, only the original (normalized) texts themselves are offered as .txt-files.
- language/domain/**annotated**: For the annotated texts, many types of information are available, ordered in subdirectories.
- language/domain/annotated/**annotations**: The annotations can be found here, ordered in subdirectories for different formats of the data.

- language/domain/annotated/**texts** and language/domain/annotated/**texts_tokenized**: The texts of the annotated corpora can be found here, with the original (normalized) texts and the (normalized) tokenized texts in different directories.
- language/domain/annotated/annotations/**sequential_annotations**: Sequential annotations always have one token per line, followed by a tab and a sequential label (more info in next section). There are empty lines between sentences.
- language/domain/annotated/annotations/**unique_annotation_lists**: Lists of all unique annotations (lowercased, lemmatized) for the entire corpus (language-domain), with one annotation per line, followed by a tab and its regarding label.

3.1.3 Versioning

Overall, there are five versions of the ACTER datasets throughout the development of the corpora where the significant updates were mainly from version 1.0 to version 1.1.

Changes version 1.0 to version 1.1. From version 1.0 to 1.1, several modifications were made to the English, French, and Dutch corpora. In the English corpus, one named entity (NE) "*com(2007) 805 final*" was removed from the corruption domain, while the wind energy domain saw the removal of two terms, "*variable pitch blades*" and "*renewable sources*", and one NE, "*skuodas*". In the French corpus, two terms, "*indélicat*" and "*loi relative à la corruption*", were removed from the corruption domain, while the equitation domain saw the removal of "*canons*" and "*équilibration*". The wind energy domain had one term added, "*systèmes mutisources-multistockages*", while four terms ("*systèmes mutisources*", "*quadrature*", "*inductance directe*", "*résistance statorique*") and 98 NEs ("*bar*", "*esk*", "*akh*", "*tht*", "*enbw*", "*rich*", "*kama*", "*man*", "*sab*", "*mer*", "*deg*", "*mor*", "*aba*", "*abo*", "*ana*", "*azm*", "*joo*", "*jen*", "*pri*", "*han*", "*ree*", "*dav*", "*cou*", "*hol*", "*sau*", "*lal*", "*lei*", "*vet*", "*pur*", "*per*", "*her*", "*hau*", "*ans*", "*slo*", "*win*", "*thi*", "*ela*", "*stem*", "*cer*", "*lav*", "*ack*", "*e.on*", "*cim*", "*luo*", "*wik*", "*ds1103*", "*fag*", "*and*", "*alm*", "*pan*", "*rap*", "*ric*", "*saa*", "*reb*", "*bor*", "*kin*", "*sem*", "*ecr*", "*fau*", "*ukt*", "*kun*", "*creg*", "*sal*", "*bou*", "*crap*", "*mog*", "*nget*", "*stu*", "*sei*", "*lec*", "*dir*", "*nor*", "*abb*", "*doh*", "*rwe*", "*mul*", "*oud*", "*bea*", "*96/92/ce*", "*gar*", "*eri*", "*cal*", "*goi*", "*ish*", "*fra*", "*cra*", "*bna*", "*ull*", "*des*", "*ips*", "*dro*", "*uct*", "*mat*", "*ds 1104*", "*mar*", "*svk*", "*bla*", "*buh*") were removed. In the Dutch corpus, "*anticorruptie-eenheid*" was added as a term to the corruption domain, while four terms were removed. Additionally, two terms were removed from both the equitation domain and the wind energy domain.

Changes version 1.1 to version 1.2. The transition from version 1.1 to 1.2 included the addition of the heart failure domain as the test domain for the TermEval shared task.

Changes version 1.2 to version 1.3. In the subsequent update to version 1.3, wrong sources in the Dutch heart failure domain were corrected and the heart failure abbreviation was changed for consistency with four-letter domain abbreviations. Additionally, a GitHub repository for the data was created and submitted to CLARIN.

Changes version 1.3 to version 1.4. The changes from version 1.3 to 1.4 involved applying limited normalization to both texts and annotations, including standardizing all dashes, single quotes, and double quotes.

Changes version 1.4 to version 1.5. The update from version 1.4 to 1.5 did not involve many changes to the actual annotations but included a significant update to their presentation. A few very long NEs were removed from the wind energy and heart failure sections, and the normalization process was updated by replacing "*İ*" with "*I*" in annotations and removing rare problematic characters. The data structure was reorganized to include sequential annotations and tokenized versions of annotations.

Our research began at the release of version 1.2, marking the inclusion of the heart failure domain as a test domain for the TermEval shared task. This foundational step

allowed us to focus on a specific context, enrich our data and methodologies, and better compare our research with other related work on the same datasets with the same settings. As we progressed, we consistently refined and enhanced our processes, culminating in the update to version 1.5. This latest version introduced significant improvements in the presentation and normalization of annotations, ensuring that our methodologies remained robust, reliable, and up-to-date with the latest standards in the field. We have maintained methodological consistency through these iterative updates, ensuring that our research remains precise and applicable.

3.1.4 License

The dataset was released as an open-source corpus on Github[1] under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)[2]. Therefore, the data can be freely used and adapted for non-commercial purposes, provided the work of Rigouts Terryn, Hoste, and Lefever (2020) is cited and any changes made to the data are clearly stated.

3.2 Slovenian Corpus of Term-annotated Texts (RSDO5)

3.2.1 Description

Designed for training automatic terminology extraction, the Slovenian Corpus of Term-annotated Texts (RSDO5)[3] corpus (Jemec Tomazin et al., 2021) provides a rich resource of manually annotated terms in Slovenian language. As part of the RSDO national project, the RSDO5 corpus was manually compiled and annotated and contains 12 documents totaling about 250,000 words from the fields of *biomechanics* (bim), *chemistry* (kem), *veterinary* (vet), and *linguistics* (ling). The data were collected from diverse sources, including Ph.D. theses (3), a Ph.D. thesis-based scientific book (1), graduate-level textbooks (4), and journal articles (4) published between 2000 and 2019.

Besides the manually annotated terms, the corpus also provided additional information with Universal Dependencies annotations, i.e., tokenization, sentence segmentation, lemmatization, morphological features, and dependency syntax. However, in our research, we only leverage the original text with the term labels, where we consider all terms and do not distinguish between in-domain and out-of-domain terms. The general statistical details of the corpus are provided in CLARIN[4].

Table 3.2: RSDO5 corpus counts (only annotated parts of corpus).

Domain	**Language**	**# files**	**# tokens**	**# terms**
Biomechanics (bim)	Slovene	3	61,344	2,319
Chemistry (kem)		3	65,012	2,409
Veterinary (vet)		3	75,182	4,748
Linguistics (ling)		3	109,050	4,601

[1] https://github.com/AylaRT/ACTER
[2] https://creativecommons.org/licenses/by-nc-sa/4.0/
[3] https://www.clarin.si/repository/xmlui/handle/11356/1470
[4] https://www.clarin.si/repository/xmlui/handle/11356/1470

3.2.2 Data Structure

Four different formats have been provided, including corpus in source Text Encoding Initiative (TEI) format, CoNLL-U format, vertical format, and plain text format. Within the scope of our research, we focus on the CoNLL-U format, whose latest version (version 1.1) has been structured as follows:

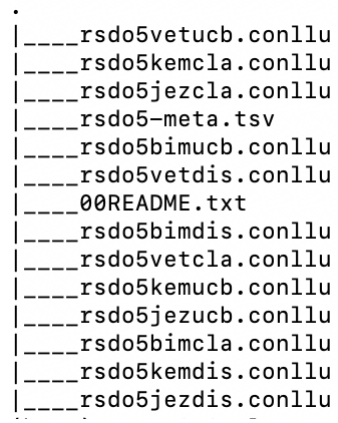


Figure 3.3: The RSDO5 CONLL-U corpus repository overview.

- **00README.txt** and **rsdo5-meta.tsv**: These files provide metadata about the corpus, including file, domain, author, title, source, publisher, date, URL, number of terms, words, and tokens.
- **rsdo5jez***, **rsdo5kem***, **rsdo5vet***, and **rsdo5bim***: These files provide identical CONLL-U format for four different domains, including Linguistics, Chemistry, Veterinary Science, and Biomechanics, respectively. The annotated terms as marked in the 10th, MISC column in IOB format. The ending **cla**, **dis**, and **ucb** represents the sources of the files (**cla** for scientific article, **dis** for scientific monograph, and **ucb** for college textbook).

3.2.3 Versioning

Overall, there are two versions of RSDO5 datasets throughout the development of the corpora. The details of the changes are defined below.

Changes version 1.0 to version 1.1. This update included enriching the existing terms within the TEI and vertical format files. This enrichment came in the form of additional markings that distinguish between terms relevant to the specific domain (in-domain) and those that are not (out-domain).

Our research deliberately targeted the CONLL-U format available in version 1.1. This decision ensured that our work remained up-to-date and applicable. By using the most recent format, we maintained the precision and broader applicability of our research findings.

3.2.4 License

Sponsored by the Ministry of Culture (C3340-20-278001) "Development of Slovene in a Digital Environment" project, the dataset was released as an open-source corpus on Clarin[5]

[5]https://www.clarin.si/repository/xmlui/handle/11356/1470

and integrated into KonText[6] platform. The corpus was also under the license of Creative Commons-Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). Therefore, the data can be freely used and adapted for our studies.

3.3 Discussion

We have summarized our description of two corpora that we used to experiment and test our hypotheses, as described in Chapters 4 to 6, including a general overview of the datasets, the data structure, different versions of the data, and the version we used in our thesis with the associated licenses. In the next three Chapters, we take a closer look at three main directions for extracting the candidate terms, including hypotheses and related methods that we apply and test in the two corpora mentioned.

[6]https://www.clarin.si/kontext/corpora/corplist

Chapter 4

Sequence-labeling Approach for ATE Tasks

This chapter focuses on the first main hypothesis of this dissertation, namely **H1: Terminology Extraction Benefits from Sequence Labeling Models** and more specifically the following five specific hypotheses concerning the claim that sequence labeling models improve terminology extraction:

- **[H1.1] Token Classification Models vs. Binary Classification Models**: *"A token classifier trained on a monolingual dataset in cross-domain setting surpasses the performance of binary classification system in extracting the candidate terms."*
- **[H1.2] Cross-lingual Transfer vs. Monolingual Learning:** *"In a zero-shot cross-lingual setting, a token classifier achieves comparable results to monolingual training in a target language."*
- **[H1.3] Multilingual Learning vs. Monolingual Learning:** *"A token classifier trained on multilingual datasets and applied to a seen target language outperforms the monolingual models trained on the target language and cross-lingual models not trained on the target language."*
- **[H1.4] The Impact of Labeled Semantics Information in Terminology Extraction:** *"The integration of label semantic information into a token classifier based on BERT outperforms the base model."*
- **[H1.5] The Impact of Mixture of Experts in Terminology Extraction:** *"A novel token classification head architecture that combines a mixture of experts (MoE) and Recurrent neural networks (RNN) on a transformer-based model outperforms the base token classification model."*

These five hypotheses are detailed in Section 4.1 when the ATE task is considered from the perspective of sequence-labeling (token classification) tasks. While hypotheses H1.4 and H1.5 are applied to different downstream NLP tasks, in the context of our dissertation, we report only on the results of terminology extraction. Meanwhile, in Section 4.2, we make a comparison among five hypotheses against the benchmark with further analysis to understand the classifier's prediction before jumping to the conclusion in Section 4.3.

4.1 Terminology Extraction as Sequence-Labeling Tasks

This section starts with preliminary studies conducted to investigate the effectiveness of this direction on the RSDO5 corpus in Section 4.1.1, followed by empirical studies of different

models from the transformer family on the ACTER and RSDO5 datasets in monolingual learning in Section 4.1.2. Next, we investigate the best token classifier from previous studies on cross-lingual and multilingual learning to determine its general applicability in less-resourced languages in Section 4.1.3. Then, in Section 4.1.4, we propose LIT, an end-to-end architecture that integrates the Encoder-Decoder mechanism of the transformers with additional information on the semantic similarity label. Finally, we propose MOSES, an architecture that integrates MoE with an RNN layer before the token classification head in section 4.1.5. The latter two architectures are applied to different token classification tasks. However, in the context of our dissertation, we report only on the results of terminology extraction.

4.1.1 Preliminary Studies

In our preliminary study on Slovenian, we leverage the RSDO5 corpus to investigate the effectiveness of token classification (so-called sequence-labeling) models for terminology extraction tasks. This focus is particularly relevant for lesser-resourced European languages, for which there is still a significant lack of benchmarks for terminology extraction compared to other downstream NLP tasks. The RSDO5 dataset, with its specific focus on Slovenian, allows us to explore this direction and contribute to closing this research gap in under-represented languages. This part resulted in the following publications:

- (H. Tran et al., 2022) **Hanh Thi Hong Tran**, Matej Martinc, Antoine Doucet, Senja Pollak. "*A Transformer-based Sequence-labeling Approach to the Slovenian Cross-domain Automatic Term Extraction*". Slovenian Conference on Language Technologies and Digital Humanities (JTDH 2022), 2022.

Overall, we evaluate the performance of XLMR, a transformer-based pre-trained model, on the terminology extraction task. We formulated the task as a supervised cross-domain sequence labeling problem and applied it to the RSDO5 dataset, which contains texts from four diverse domains. We show that the proposed cross-domain approach surpasses the current SOTA (Ljubešić et al., 2019) for all the combinations of training and testing domains with which we experimented, thus establishing a new SOTA for the terminology extraction task in the Slovenian corpus.

4.1.1.1 Task Formulation

Let:

- $X = \{Tok_1, Tok_2, ..., Tok_n\}$ be a sequence of n tokens in a text document.
- $T = \{term_1, term_2, ..., term_m\}$ be a set of m terms from a gold standard list.
- $L = \{B, I, O\}$ be the label set for the BIO annotation scheme, where B is the beginning of a term, I is inside a term, and O is outside of a term.
- $f(Tok_i)$ be the pre-trained XLMR model's output vector for token Tok_i.

Objective:
The goal is to learn a function $g : X \rightarrow L^n$ that assigns a label from L to each token in the sequence X.
Model:
The architecture consists of the following components:

1. **Embedding layer**: This layer maps each token Tok_i to a dense vector representation E_i.

2. **Encoder layer**: This pre-trained transformer-based model processes the sequence of encoded tokens ($E = \{E_1, E_2, ..., E_n\}$) to capture contextual information. We can denote the output of the Encoder for token i as T_i.

3. **Classification head**: This layer takes the encoded representation T_i for each token and predicts the label probability distribution:

$$p(l|T_i) = softmax(W_c * T_i + b_c) \tag{4.1}$$

where W_c is the weight matrix of the classification head, b_c is the bias vector of the classification head, and $p(l|T_i)$ is the probability of label l being assigned to token i.

The general workflow of the model is visualized in Figure 4.1.

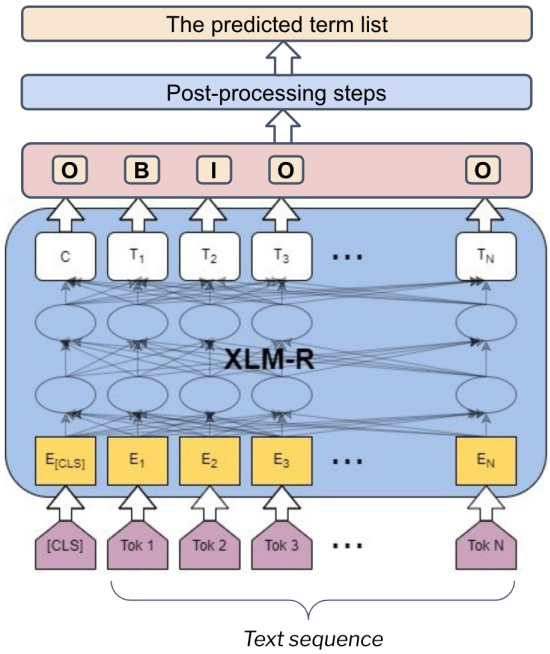


Figure 4.1: The overall XLMR architecture.

Training:
The model is trained on a labeled dataset where each token Tok_i has a corresponding gold-standard label $l_i \in L$. The training objective is to minimize the loss function measuring the discrepancy between the predicted and true labels. A common loss function for sequence labeling tasks is the cross-entropy loss:

$$\text{Loss} = -\sum_{i=1}^{n}\sum_{l \in L} y_i \cdot \log(p(l|z_i)) \tag{4.2}$$

The predictions are then fed into a post-processing step to extract the final candidate terms in the form of a list, which is in the same format as the original gold standard.

4.1.1.2 Architecture

We consider ATE as a token classification task where the model returns a label for each token in a text sequence. We use the BIO labeling mechanism (Lang et al., 2021; Rigouts Terryn et al., 2021). The terms from the gold standard list are first assigned to the tokens in the raw text. Each word inside the text sequence is annotated with one of the three labels (see examples in Figure 4.2).

Figure 4.2: An example of the BIO mechanism on a text sequence from Slovenian corpus.

We experiment with XLMR[1] (Conneau et al., 2019), a transformer-based model pre-trained on 2.5 TB of filtered CommonCrawl data with 100 languages. With the proliferation of non-English models (e.g., CamemBERT for French, Finnish BERT, German BERT), XLMR, the multilingual version of RoBERTa (Y. Liu et al., 2019), is a generic cross-lingual sentence Encoder that achieves benchmark performance on multiple downstream NLP tasks, including ATE for rich-resourced languages (e.g., English) (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020). Based on this well-documented SOTA performance on several related tasks, we opt to employ XLMR in a monolingual setting on our low-resourced Slovenian corpus (in comparison with SloBERTa later in Chapter 4.1.2).

The overall architecture is presented in Figure 4.1. First, we divide the dataset into train-validation-test splits. The train split is used for fine-tuning the pre-trained language model. The validation split is applied to prevent over-fitting during the fine-tuning phase. Finally, the test split, which is not adopted during training, is used to evaluate the method. The model is fine-tuned on the training set to predict for each word in the sequence the probability of whether it is a part of the term (B, I) or not (O). For this purpose, an additional token classification head containing a feed-forward layer with a softmax activation is added on top of the model.

4.1.1.3 Experimental Setup

We investigate the effectiveness of cross-domain learning, testing the transfer of knowledge from one domain to another, and thus evaluating the model's capability to extract terms in new unseen domains. Therefore, we fine-tune the model in two domains (e.g., *biomechanics* and *chemistry*), validate it in the third domain (e.g., *veterinary*), and test it in the fourth domain that is not included in the training set (e.g., *linguistics*). We consider terminology extraction as a sequence-labeling or token classification task with a BIO annotation scheme.

We employ the XLMR token classification model and its "fast" XLMR tokenizer from the Huggingface library[2]. We fine-tune the model for up to 20 epochs (i.e., we employ the early stopping regime) using the learning rate of 2e-05, training and evaluation batch size of 32, and sequence length of 512 tokens, since this hyperparameter configuration had the best performance in the validation set. Documents containing more than 512 tokens are truncated, while documents with fewer than 512 tokens are padded with a special $< PAD >$ token at the end. During fine-tuning, the model is evaluated on the validation set after each training epoch, and the model with the best performance is applied to the test set.

[1] https://huggingface.co/FacebookAI/xlm-roberta-base
[2] https://huggingface.co/models

For each word in a word sequence, the model predicts whether it is part of a term (B, I) or not (O). The sequences identified as terms are extracted from the text and included in a set of all predicted candidate terms. In a post-processing step, all candidate terms are lowercase before we compare our derived candidate list with the gold standard using the standard evaluation metrics (precision, recall, and F_1).

4.1.2 Empirical Studies of Transformer-based Models

Inspired by the performance of the token classifier described in Section 4.1.1, we propose an empirical evaluation of several transformers-based language models, pre-trained on either monolingual or multilingual corpora, including both masked (e.g., BERT, RoBERTa) and autoregressive (e.g., XLNet) models used in the cross-domain ATE tasks, which aligns with hypothesis H1.1. In addition, we not only fill the research gap in the Slovenian ATE task by experimenting with different models to achieve a new SOTA in the RSDO5 corpus, but we also elaborate our studies on different languages from ACTER corpora to evaluate the consistency of the performance of different dominant and lesser-represented European languages. Finally, we propose a simple but efficient late-fusion approach that further improves SOTA in this field. This part resulted in the following publication:

- (H. T. H. Tran, Martinc, Pelicon, et al., 2022) **Hanh Thi Hong Tran**, Matej Martinc, Andraz Repar, Antoine Doucet, Senja Pollak. "*Ensembling Transformers for Cross-domain Automatic Term Extraction*". International Conference on Asian Digital Libraries (ICADL 2022). Cham: Springer International Publishing, 2022.

4.1.2.1 Task Formulation

Evaluation of Individual Models:

- Let $M = \{M_1, M_2, ..., M_n\}$ be a set of n transformer-based models evaluated for the ATE task. M_i can be a multilingual model or a monolingual model.
- For each model M_i, we can define an evaluation metric (e.g., precision, recall, F_1) as $E_i(M_i, D_{test})$, where:
  - E_i represents the evaluation metric.
  - D_{test} is the test dataset used for evaluation.

Ensemble Approach:

The ensemble or late fusion approach combines the results of the two models that perform best in the following three settings to potentially improve performance: [1] the best monolingual and multilingual models; [2] the two best monolingual models; and [3] the two best multilingual models. Here, two strategies are used:

1. **Union**: This strategy combines the list of candidate terms from the two models that perform the best in all three settings. Mathematically, we can represent the union operation as:
$$U = \bigcup (T_i \mid M_i \in M) \tag{4.3}$$
where:
   - U is the combined candidate term list.
   - T_i is the candidate term list generated by model M_i.

2. **Intersection**: This strategy takes only the terms identified by the two highest-performing models in the three settings. Mathematically, we can represent the intersection operation as:

$$I = \bigcap (T_i \mid M_i \in M) \tag{4.4}$$

where:

- I is the intersected candidate term list.

Overall Evaluation:

The final evaluation metric, the F_1-score, can be calculated based on the chosen ensemble approach (U or I) for each combination and compared with the performance of the individual models (E_i).

4.1.2.2 Architecture

We conduct a systematic evaluation of transformer-based models that were pre-trained on monolingual and multilingual corpora for the ATE task modeled as sequence labeling. The models are obtained from Huggingface according to the number of downloads and likes. The chosen models are presented in Figure 4.3.

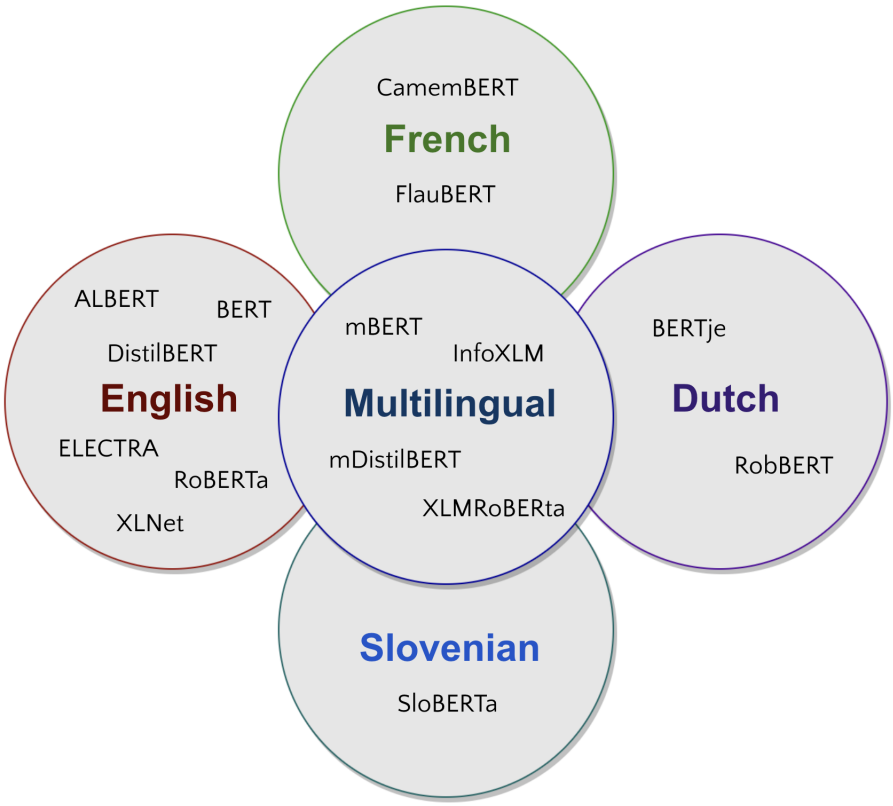


Figure 4.3: Empirical evaluation of pre-trained language models on the ATE task.

Regarding multilingual pre-trained systems, we investigate the performance of mBERT[3] (Devlin et al., 2018), mDistilBERT[4] (Sanh et al., 2019), InforXLM[5] (Chi et al., 2021), and

[3]https://huggingface.co/google-bert/bert-base-multilingual-uncased
[4]https://huggingface.co/distilbert/distilbert-base-multilingual-cased
[5]https://huggingface.co/microsoft/infoxlm-base

XLMR[6] (so-called XLMRoBERTa) (Conneau et al., 2019). All the chosen multilingual models are fine-tuned in a monolingual fashion due to findings from related work (Lang et al., 2021) showing that no (or only marginal) gains are obtained if the model is fine-tuned on the multilingual training data.

Regarding the monolingual pre-trained models, we evaluate several English autoencoding Transformer variants, including ALBERT[7] (Lan et al., 2019), BERT[8] (Devlin et al., 2018), DistilBERT[9] (Sanh et al., 2019), ELECTRA[10] and RoBERTa[11] (Y. Liu et al., 2019), and one autoregressive model, XLNet[12] (Z. Yang et al., 2019). For French, we use CamemBERT[13] (Martin et al., 2019) and FlauBERT[14] (H. Le et al., 2020). For Dutch, we employ BERTje[15] and RobBERT[16]. For Slovenian, we choose SloBERTa[17], a RoBERTa-based model trained on a large Slovenian corpus.

Based on the results of the above empirical studies, we propose a novel approach to late fusion for ATE tasks. This decision is guided by the general tendency that precision is better than recall for all tested monolingual and multilingual models. This leads us to believe that by combining the results of different models, we can achieve improvements in recall and hence overall F_1. We consider two strategies to combine the outputs of the different models of the ensemble, namely the union and the overlap of the lists of candidate terms from the models of the ensemble. See the whole procedure in Figure 4.4.

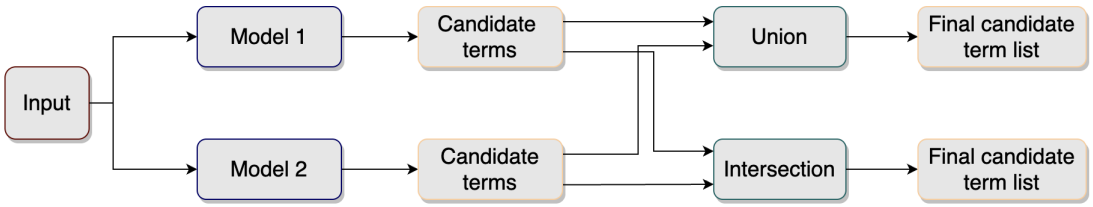


Figure 4.4: Empirical evaluation of pre-trained language models on the ATE task.

We hypothesize that by combining the outputs of the two models, we will be able to significantly improve the recall of the terminology extraction system. To validate this hypothesis, we test three combinations, namely, we combine the outputs of the [1] best monolingual and multilingual models, [2] two best monolingual models, and [3] two best multilingual models.

4.1.2.3 Experimental Setup

The experiments were performed with two datasets, RSDO5 v1.1 (Jemec Tomazin et al., 2021) and ACTER v1.5 (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020), due to their diverse domains and well-documented guidelines. In the RSDO5 v1.1 corpus, we share the same configuration setup as in Chapter 4.1.1. We experiment with 12 different combinations of training, validation, and testing data, where two domains are used for training, a

[6] https://huggingface.co/FacebookAI/xlm-roberta-base
[7] https://huggingface.co/albert/albert-base-v1 and https://huggingface.co/albert/albert-base-v2
[8] https://huggingface.co/google-bert/bert-base-uncased
[9] https://huggingface.co/distilbert/distilbert-base-uncased
[10] https://huggingface.co/google/electra-small-generator
[11] https://huggingface.co/https://huggingface.co/FacebookAI/xlm-roberta-base
[12] https://huggingface.co/xlnet/xlnet-base-cased
[13] https://huggingface.co/almanach/camembert-base
[14] https://huggingface.co/flaubert/flaubert_base_uncased
[15] https://huggingface.co/GroNLP/bert-base-dutch-cased
[16] https://huggingface.co/pdelobelle/robbert-v2-dutch-base
[17] https://huggingface.co/EMBEDDIA/sloberta

third one for validation, and a fourth one for testing. Meanwhile, for all three languages in the ACTER v1.5 corpora, we consider *corruption* and *wind energy* for training, *equitation* for validation, and *heart failure* as a test set, same as in the TermEval 2020 shared task (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020).

For both datasets, we consider terminology extraction as a sequence-labeling task with a BIO annotation scheme. The chosen model predicts for each word in a word sequence whether it is a part of a term (B, I) or not (O). The extracted candidate terms are then formulated as a set and converted to lowercase in a post-processing step. Only then do we compare them with the gold standard using the evaluation metrics (precision, recall, and F_1).

4.1.3 Cross-lingual and Multilingual Learning

Inspired by the success of transformer-based models as (binary) sequence classifiers in the TermEval 2020 competition (Hazem et al., 2020), in our adaptation as a token classifier (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022; H. Tran et al., 2022), and the emergence of cross-lingual learning (Lang et al., 2021), we propose to further explore the performance of the XLMR (Conneau et al., 2020) pre-trained model in cross-lingual learning, where the model is fine-tuned in one or more languages and tested in a new unseen language; and in multilingual learning, where the model is fine-tuned on several languages and tested in a seen language. The choice of pre-trained models was based on two reasons: [1] the results of the empirical studies in Chapter 4.1.3; [2] the multilingualism of XLMR where the pre-trained model covers all four languages in the training set.

This part resulted in the following publication:

- (H. T. H. Tran, Martinc, et al., 2024) **Hanh Thi Hong Tran**, Matej Martinc, Andraz Repar, Nikola Ljubesic, Antoine Doucet, Senja Pollak. *Can Cross-domain Term Extraction Benefit from Cross-lingual Transfer and Nested Term Labeling?* Special Issue of Discovery Science. Machine Learning. 2024.

- (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022) **Hanh Thi Hong Tran**, Matej Martinc, Antoine Doucet, Senja Pollak. "*Can Cross-domain Term Extraction Benefit from the Cross-lingual Transfer?.* International Conference on Discovery Science (DS 2022). Cham: Springer Nature Switzerland, 2022.

Overall, we model terminology extraction as a sequence-labeling task and systematically evaluate the performance of our token classifier (e.g., XLMR) in cross-lingual settings using the ACTER v1.5 and in multilingual settings for both ACTER v1.5 and RSDO5 v1.1 corpora. We compare the performance of cross-lingual and multilingual learning with that of monolingual learning to determine its general applicability in less well-resourced languages. This aligns with hypotheses H1.2 and H1.3.

4.1.3.1 Task Formulation

Let:

- $L = \{English, French, Dutch, Slovenian\}$ be the set of four languages we investigate.

- k be an integer representing the number of languages considered for training (1 for monolingual, 2 or higher for cross-lingual).

- C_k be the collection of possible tuples of size k that can be constructed from languages in L. For example, C_2 denotes the collection of all possible two-language combinations in a set of four languages, e.g.

$$C_2 = \{(English, French), (English, Dutch), ...\} \tag{4.5}$$

Cardinality of Language Combinations:

We denote the i^{th} tuple of size k with C_k^i, e.g., for the previous example, C_2^1 would yield $(English, French)$. The cardinality of the collection C_k, $|C_k|$ is formulated as:

$$|C_k| = \begin{pmatrix} 4 \\ k \end{pmatrix} \tag{4.6}$$

It represents the number of ways to choose k distinct elements (languages) from a set of 4 (L) without considering the order.

Training set:

We create i^{th} training dataset D from the collection of tuples C_k of size k, as a concatenation of datasets in the tuple, or more formally $D_{i,k}$:

$$D_{i,k} = \bigcup_{language \in C_k^i} train\_split(language) \tag{4.7}$$

where *train-split* represents the respective data-split of the given language.

Monolingual learning ($k = 1, |C_1|$)

- $C_1 = L$: In this case, C_1 contains all four individual languages (English, French, Dutch, Slovenian) for monolingual training, respectively.

Cross-lingual learning ($k > 1, D_{i,k}$ for $i \in [1, |C_4|]$)

- This specifies the specific combination of $k - 1$ languages for training in the i^{th} instance and tests on new unseen k language.

Multilingual learning ($k = 4, D_{i,4}$ for $i \in [1, |C_4|]$)

- C_k^i This represents the i^{th} tuple of size k in the collection C_k. It specifies the specific combination of k languages for training in the i^{th} instance.

4.1.3.2 Architecture

We apply the cross-domain performance of the XLMR token classifier in cross-lingual and multilingual learning compared to monolingual learning. Altogether, different scenarios are tested and described below.

1. **Monolingual setup.** We evaluate how well the model performs when a language-specific training corpus is available and there is a match between the language of the train set and the language of the test set. To allow a better comparison with other existing approaches, we apply the same configuration as in the TermEval 2020 share task where *heart failure* of each language is considered as a test set. This means that we train three monolingual models for three languages (English, French, and Dutch) and test each model in the same language for each annotation regime. We also train 12 monolingual models for each annotation regime for Slovenian, using 12 different combinations of the split of training, validation, and testing for the domains.

2. **Cross-lingual setup.** We evaluate the model's ability to apply knowledge learned in one or more languages to terminology extraction in another unseen language. Therefore, we fine-tune the model in one or more languages (e.g., English and Dutch) and test it in another language that is not included in the training set (e.g., French). In this scenario, we thus investigate how well the model works without the language-specific training corpus and how well the knowledge transfer between different languages is.

3. **Multilingual setup.** To fine-tune our model, we use [1] training datasets for all languages in the ACTER dataset (e.g., English, French, and Dutch) or [2] training datasets for all languages in the ACTER dataset and the Slovenian training dataset from the RSDO5 corpus and then apply the model to the test sets of all languages. In this way, we investigate whether adding more data from other languages to the training dataset corresponding to the target language improves the predictive performance of the model.

All settings are applied in a cross-domain evaluation scenario, where we use two domains for training, another domain for validation, and the rest for testing. An exception is the multilingual and cross-lingual settings with the additional Slovenian corpus in the training set, where we use two domains from ACTER corpora and all domains from the RSDO5 corpus for training to predict the ACTER test set and vice versa for RSDO5 corpus. In this way, we can evaluate the generalization capabilities of the model to adapt the knowledge in one or more domains to a new, unseen, arbitrary domain, which is much more useful.

4.1.3.3 Experimental Setup

Monolingual learning shares the same experimental setup with the XLMR classifier described for the methods in Chapter 4.1.1. Meanwhile, cross-lingual and multilingual learning share the same model configuration and the data train-validation-test split setup for both datasets.

4.1.4 Label-Informed Transformers (LIT) Models

Transformers-based language models have led to the investigation of various embedding and modeling techniques for several downstream NLP tasks. Nevertheless, the comprehensive exploration of semantic information about the label from the Encoder and Decoder components has not yet been fully realized in these tasks. Thus, aligning with hypothesis H1.4, we propose LIT, an end-to-end pipeline architecture that integrates the transformer's Encoder-Decoder mechanism with an additional semantic similarity label for token classification tasks. This part resulted in the following publication:

- (Accepted) (Sun et al., 2024) Wenjun Sun, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Mickaël Coustaty and Antoine Doucet. "*LIT: Label-Informed Transformers on Token-based Classification*". 28^{th} International Conference on Theory and Practice of Digital Libraries (TPDL 2024), 2024.

Within the scope of this dissertation, we only report the results on the ATE task, which was one of the downstream tasks tested.

4.1.4.1 Architecture

The LIT architecture is shown in Figure 4.5. It consists of a backbone language model, an Encoder, a feature extractor, a Decoder, and a cosine similarity operation. The first three

are used to compute the embedding of the target token, the Decoder is used to obtain the embedding of each label, and finally, the cosine similarity is used to obtain the final prediction.

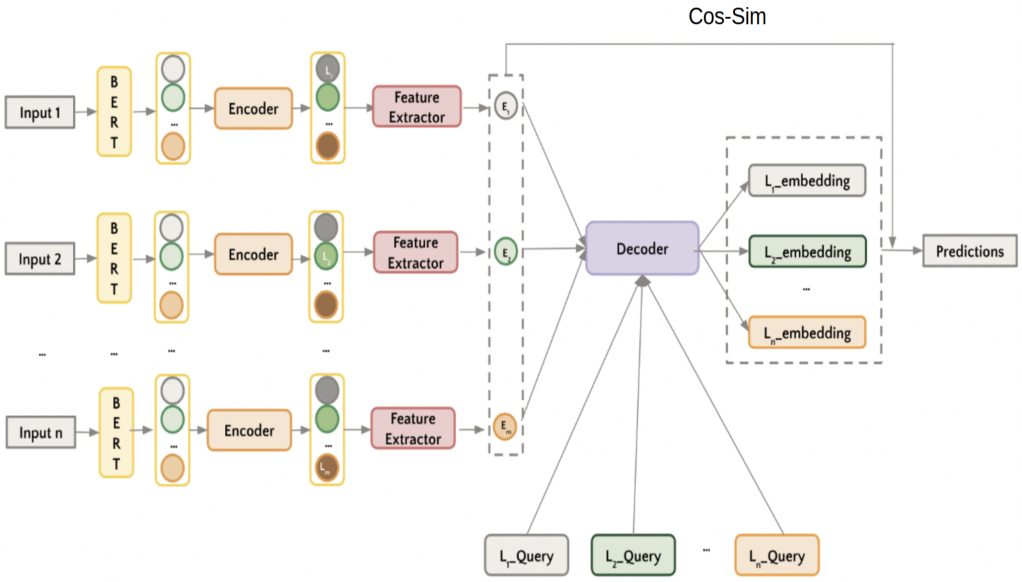


Figure 4.5: General architecture of LIT approach.

First, when the training data are loaded, the input is represented by a sequence of tokens using BERT embeddings. Then, the vector representation of each token is obtained by the language model and fed directly to the Encoder for further computation. We use this Encoder to process the semantics (e.g., domain-specific terms). During the tokenization process, some tokens are divided into multiple sub-tokens, we thus adopt the vector representation of the first sub-token as the overall feature of this token. Then, the average vector value of each target token is extracted as the input of the Decoder. Mathematically speaking, given the original sequence $\{T_1, T_2, ..., T_x\}$, the tokenization process returns the output sequence of $\{T_{1_1}, T_{1_2}, ..., T_{x_y}\}$, where T_{x_y} means the y^{th} sub-token of the x^{th} token. After BERT embeddings and the Encoder, the resulting token embeddings are $\{Em_{1_1}, Em_{1_2}, ..., Em_{x_y}\}$, where Em_{x_y} refers to the embedding of the y^{th} sub-token of the x^{th} token and $L(Em)$ is the embedding of each label of Em_x, e.g., the representation of $TERM$, given 1 being the first sub-token, can be formulated as:

$$TERM = Avg(1_{\{L(Em_{i_1})==TERM\}} Em_{i_1}); i = \{0, \ldots, x\} \tag{4.8}$$

The feature extractor is used to calculate the average embedding of the term token in sentences. Then, sequences are grouped into corresponding input groups according to the labels they contain (e.g., a label for terminology extraction and five labels for the Historical NER tasks). If one sequence contains multiple labels, it will be organized into multiple corresponding groups. This is to allow the model to learn the semantics of all labels during training. These representations are fed into the Decoder as the representation along with the n decoding vectors as another input to the Decoder according to the number of labels. Finally, the n outputs of the Decoder are obtained and used as the representation of each label. The predicted token label is the one that maximizes the cosine similarity (Cos_sim) between the Encoder and Decoder outputs:

$$L(Em_{x_{s1}}) = Max(Cos\_sim(Em_{x_{s1}}, Label))) \tag{4.9}$$

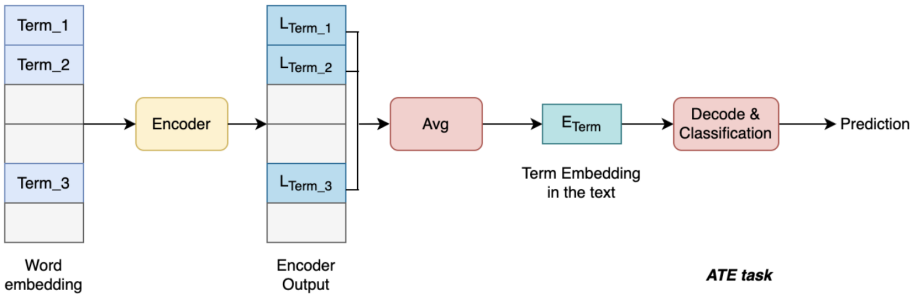


Figure 4.6: Architecture adaptation on terminology extraction as the token classification task.

where *Label* means the embeddings of labels. This mechanism is shown in Figure 4.6 for both token-based classification tasks. During inference, the feature extractor is not needed, as the final result is obtained by comparing the output of the Encoder with the embedding of the label.

4.1.4.2 Experimental Setup

To make long texts compatible with the BERT model, we set a window length and the truncation stride to 100 and use the cross-entropy function for the loss calculation. For the language model, we fine-tuned BERT based on the dataset and our specific task (for the terminology extraction task, we used the cased version of BERT[18]). Our model's Encoder and Decoder are both in the transformer's structure.

4.1.5 Mixture of Specialized Experts (MOSES) for Supervised Extraction

In recent years, foundation models have completely revolutionized the way NLP is done and the development of LLMs has enabled task-agnostic architectures that can solve downstream NLP tasks in a zero-shot fashion, with no labeled data required (Brown et al., 2020). With the rise of LLMs, novel training regimes and architectural components have been proposed, such as low-rank adaptation (LoRA) (Hu et al., 2021) and the mixture of experts (MoE) (Jacobs et al., 1991), which make these models more efficient and training less computationally expensive and faster. While these components have been successfully employed in an unsupervised setting with abundant data, studies that would research their effectiveness in a supervised setting with fewer resources are scarce. The main reason for this is that these components have been developed with the specific aim of improving the scalability of models (Shazeer et al., 2017) and therefore their impact on the performance of models is somewhat neglected in most studies or only measured indirectly, i.e., the studies focus on how the use of these components enables greater scalability, which in turn leads to performance gains.

In contrast, our studies focus on the performance improvements we can obtain from one of these components without increasing the size of the backbone model. More specifically, we are interested in how MoE can be used to improve several text extraction tasks in a supervised setting with far fewer resources than is typically the case in an unsupervised language modeling setting. The general hypothesis is that, the same as for text generation, it is beneficial for text extraction to allow specific parts of the network to specialize for

[18]https://huggingface.co/bert-base-cased

specific types of tokens. More specifically, we investigate whether the introduction of a gating network that assigns different parts of the sequence to specialized layers that only take care of certain subsequences and tokens according to their semantic and grammatical role could improve SOTA. As a result, we test our approach on several sequence-labeling tasks with different amounts of available training data to determine the data threshold that still allows the convergence of expert layers.

This part aligns with hypothesis H1.5 and results in the following publication under evaluation:

- (Under review) Matej Martinc, **Hanh Thi Hong Tran**, Boskho Koloski, Senja Pollak. "*MOSES: Mixture of Specialized Experts for Supervised Extraction*". Transactions of the Association for Computational Linguistics (TACL 2024), 2024.

In the scope of our dissertation, we only report the architecture and results for ATE tasks.

4.1.5.1 Architecture

MOSES (visualized in Figure 4.7) is based on fine-tuning a pre-trained transformer architecture with a customized token classification head. As a backbone model, we employ DeBERTa (He et al., 2020) for the English corpus and mDeBERTa for French and Dutch corpora of ACTER datasets. It was chosen since it showcased SOTA performance on several downstream NLP tasks, among them also the majority of three sequence labeling tasks that we investigate, such as NER (Shon et al., 2022), keyword extraction, and our specific terminology extraction tasks.

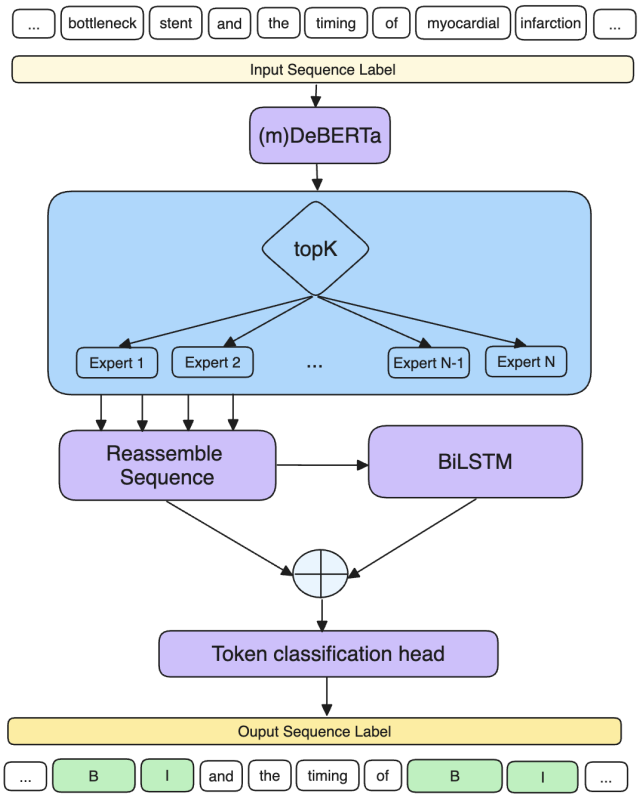


Figure 4.7: The MOSES general architecture given the terminology extraction input.

The main novelty of our approach is a customized token classification head that contains several components. The output logits from the (m)DeBERTa backbone are first fed to the gate network or router, which determines which tokens in the input sequence are sent to which expert. More specifically, in the setting with n experts the gate network (G) decides which experts (E) receive a part of the input:

$$y = \sum_{i=1}^{n} G(x)_i E_i(x)$$

We model the routing as a weighted multiplication, where hypothetically all experts could contribute to processing each part of the input. Nevertheless, the additional top_k parameter ensures that only the top k experts contribute to the respective part of the input, and other experts are excluded from the computational graph, thus saving computational resources. Following Shazeer et al. (2017), we employ the so-called Noisy Top-k Gating, which introduces some (tunable) noise and then keeps the top k values. Noise is introduced for load balancing, i.e., to prevent the gating network from converging in a way that the same few experts are activated for most of the tokens, which would make training inefficient. More specifically, the final weight distribution is obtained by feeding the input (x) through a dense layer (W_g) and adding some random noise in the following way:

$$H(x)_i = (x \cdot W_g)_i + \text{StandardNormal}() \cdot \text{Softplus}((x \cdot W_{noise})_i)$$

After that, we filter out only the top k experts out of v:

$$KeepTopK(v, k)_i = \{v_i \text{ if } v_i \text{ in top k, otherwise} - \inf\}$$

We finally apply softmax only to the filtered experts:

$$G(x) = \text{Softmax}(\text{KeepTopK}(H(x), k))$$

Note that the router is composed of learnable parameters and is fine-tuned at the same time as the rest of the network.

On the other hand, each expert is represented as a neural network consisting of three sequential dense layers W_g with additional activation and dropout layers. More specifically, the expert is represented as:

$$E(x) = \text{Dropout}(\text{ReLU}(x \cdot W_g) \cdot W_g) \cdot W_g$$

The outputs of different experts, which attend to specific tokens assigned to them by the routing network, are first reassembled into a single sequence representation. After that, a two-layer randomly initialized Encoder, consisting of dropout and two recurrent neural networks (RNN) layers, is added (with element-wise summation) to this sequence. The initial motivation behind this adaptation is related to the findings from related work that suggest that recurrent layers are quite successful in modeling the positional importance of tokens in the keyword detection task (Meng et al., 2017; X. Yuan et al., 2020) and by the study of Sahrawat et al. (2020), who also reported good results when an RNN classifier and contextual embeddings generated by transformer architectures were used for keyword detection. Also, the results of the initial experiments suggested that some performance gains can be achieved by employing this modification, especially on smaller datasets.

The output sequence of the MoE and RNN Encoders is finally fed to the feed-forward classification layer, which returns the output matrix of size SL $*$ NC, where SL stands for sequence length and NC stands for the number of classes. In our case, NC is always 3, since we model all tasks as sequence labeling with the BIO annotation regime.

For each of the tasks, we perform an additional post-processing step. For example, in terminology extraction, the extracted candidate terms go through an additional filtering step to lowercase them and remove unnecessary duplication. All candidate terms at the sentence level are then added to the final list of candidate terms at the corpus level.

4.1.5.2 Experimental Setup

We employ MOSES for three different sequence-labeling tasks, namely keyword extraction, terminology extraction, and named entity recognition (NER), and test various settings to determine the effect that the proposed architectural additions have on the performance of the model. We are interested in the specific contributions that the MoE and RNN layers have on the performance of the model (either separately or together). Therefore we compare the settings containing these components to the baseline (m)DeBERTa model with a usual (dense) token classification head. However, in the scope of this dissertation, we report mainly on terminology extraction tasks.

During fine-tuning, we use low-rank adaptation (LoRA) (Hu et al., 2021) instead of fine-tuning the entire model. More specifically, we freeze all layers in the backbone (m)DeBERTa model and train only low-rank perturbations to query and value weight matrices in the model. Additionally, we train all matrices in the custom token classification head (i.e., MoE, RNN, and dense classification layers are randomly initialized and fine-tuned). Each model was fine-tuned for a maximum of 20 epochs and after each epoch, the trained model was tested on the documents chosen for hyperparameter optimization and test set model selection. The model that showed the best performance was used on the test set. The hyperparameter values used during fine-tuning include a learning rate of 2e-4, a sequence length of 512, LoRA r of 16, LoRa alpha of 16, and LoRa dropout of 0.1.

4.2 Results

In this section, we define the evaluation metrics in Section 4.2.1 to measure the performance of different approaches introduced earlier in Chapter 4 compared to the baseline as described in Chapter 4.2.2. We then present general observations and discussions on the results obtained in Subsection 4.2.3. A more detailed error analysis is provided in Section 4.2.4 to gain deeper insights into model behavior.

4.2.1 Evaluation Metrics

We evaluate each terminology extraction system by comparing the aggregated list of candidate terms extracted on the level of the whole test set with the manually annotated gold standard term list using precision (see equation 2.1), recall (see equation 2.2), and F_1-score (F_1) (see equation 2.3). These evaluation metrics have also been used in related work (Lang et al., 2021; Nugumanova et al., 2022), including the TermEval 2020 shared task (Hazem et al., 2020; Rigouts Terryn, Hoste, Drouin, & Lefever, 2020).

4.2.2 Baselines

We compared our studies with different benchmarks for each corpus. For the ACTER corpora, we included [1] the winning solutions for each language in ACTER from the TermEval 2020 shared task where the dataset was published and [2] the later proposed solutions up to the start of our research. However, for the RSDO5 corpus, due to its novelty, we compared our work with the latest studies until we started our research.

4.2.2.1 ACTER Corpora

During the TermEval 2020 shared task, we consider winning solutions from TALN-LS2N (Hazem et al., 2020) for English and French and NLPLab_UQAM (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020) for Dutch as the baselines:

- TALN-LS2N (Hazem et al., 2020) was the winning solution for English and French corpora where they used BERT as a binary classifier for terminology extraction. Given the input being the concatenation of a sentence and a selected n-gram within the sentence, the classifier labeled the n-gram as a positive training example if it is a term. Otherwise, the classifier labeled it as a negative example.

- NLPLab_UQAM (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020) was the winning solution for the Dutch corpus where they applied a BiLSTM to the task given the input being the sequence represented by pre-trained Glove word embedding.

After the TermEval 2020 shared task, several methods have been implemented in both neural and non-neural directions to surpass the baselines above, including:

- HAMLET (Rigouts Terryn et al., 2021) proposed a hybrid adaptable machine learning system to extract the candidate terms in a three-step procedure: [1] preprocessing and candidate term selection based on PoS, [2] extracting 177 different features (e.g., shape, linguistics, frequency, statistical, etc), and training on different ML classifiers (e.g., DTs, RF, multi-layer perception, logistic regression). We reported only the optimal combinations for the best terminology extraction performance.

- NMT (Lang et al., 2021) considered terminology extraction from three perspectives: [1] as a sequence classification task, [2] as a token classification task, and [3] as a neural machine translation (NMT) task. The results show that the best performance when considering the second perspective is achieved with the XLMR classifier, which serves as a benchmark in our studies.

- NMF (Nugumanova et al., 2022) combined the probabilistic topic modeling (PTM) and non-negative matrix factorization (NMF) to extract the candidate terms. Five different NMF algorithms and four different NMF initializations were tested by changing the number of topics extracted from the documents and the number of most probable words extracted from the topics. Again, we only specified the optimal combinations for the best terminology extraction performance.

All mentioned neural methods followed a cross-domain setting, where two domains (i.e., *corruption*, *wind energy*) were used for training, another domain (i.e., *equitation*) was used for validation, and the remaining domain (i.e., *heart failure*) was used for the testing phase. We shared the same configuration of train-validation-test splitting as them to make our work transparent and comparable.

4.2.2.2 RSDO5 Corpus

There is a limited amount of research available on RSDO5. Two of the most recent works on the tasks were considered as the baseline, including:

- KAS-term (Ljubešić et al., 2019) proposed supervised learning experiments (e.g., SVM with RBF kernel) that can be adapted to RSDO5 corpus to extract candidate terms from Slovene academic texts using frequency and different co-occurrence statistics (e.g., chi-square, Dice, pointwise mutual information, t-score, tf-idf, C-value).

- TermoUD (Marciniak et al., 2023) was a language-independent terminology extraction tool with a two-step pipeline: [1] Identifying the candidate phrase using predefined rules, and [2] ranking the candidate terms based on C-value.

4.2.3 Quantitative Results

We report the performance of our methods for each of the five hypotheses described above on two corpora using precision, recall, and F_1-score as evaluation metrics.

4.2.3.1 ACTER Corpora

Tables 4.1, 4.2, and 4.3 reported the performance of our approaches in English, French, and Dutch corpora (ACTER version 1.5), respectively, compared with the baseline (Group 5) described in Section 4.2.2. Our methods include four groups: [1] different transformer-based models in monolingual learning (Group 1); [2] XLMR in cross-lingual and multilingual learning (Group 2); [3] LIT architecture (Group 3); and [4] MOSES architecture (Group 4).

Transformer-based models as Token Classifiers vs. Baselines

We compared our empirical experiments using different transformer-based models as token classifiers fine-tuned on monolingual datasets (Group 1) with the four baselines for each language (Group 5). While the monolingual pre-trained models were competitive in English (e.g., RoBERTa), and French (e.g., CamemBERT) corpora, the multilingual models (e.g., XLMR) proved to outperform both versions of the Dutch corpus. This highlighted the impact of monolingual pre-trained models on the test set of their specific languages and the potential of multilingual models to tackle the dataset with fewer resources compared to the dominant ones (e.g., English). Moreover, our approaches using BERT as a token classifier outperformed the approaches of TALN-LS2N where BERT was considered as a sequence classifier for both versions of the gold standard in English and French, except for the English ANN version when using the same BERT as a backbone. To be precise, there was an increase of 6.9 percentage points in the English NES version, 2.5 percentage points in the French ANN version, and 9.3 percentage points in the Dutch NES version, while there was only a decrease of 2 percentage points in the English ANN version when BERT was used as the token classifier for the task. Furthermore, most of the other transformers' token classifiers outperformed the sequence classifier of the TALN-LS2N baseline. Thus, our hypothesis H1.1 was confirmed: *"A token classifier trained on a monolingual dataset in cross-domain setting surpasses the performance of the binary classification system in extracting candidate terms."*.

Note that in our empirical studies, monolingual models (e.g., CamemBERT) performed better on a specific language test set (e.g., French test set) than other models pre-trained in that particular language (e.g., French) or fine-tuned models in cross-lingual or multilingual learning due to several factors. First, monolingual pre-trained models are trained exclusively on one language, allowing them to specialize and optimize the linguistic structures, nuances, and vocabulary of the language. In contrast, multilingual pre-trained models must balance the learning of multiple languages, which can lead to less effective optimization for a single language. Second, the former models may have a vocabulary and tokenization scheme specifically tailored to French, better capturing its unique morphological and syntactic features. The latter models, on the other hand, use common vocabularies that are less efficient in capturing the specificities of each language, especially for languages with rich morphology or specific characters. Despite the performance of the monolingual model in that particular language, it often provides less competitive performance when it

Table 4.1: The evaluation of different approaches in the English version of the ACTER corpora. The best result for each evaluation metric appears in bold for each version.

	Settings	ANN version			NES version		
		P	**R**	**F_1**	**P**	**R**	**F_1**
	(Group 1) Transformer-based models in Monolingual Learning						
Mono	$ALBERT_{v1}$	52.6	47.4	49.9	54.4	54.6	54.5
	$ALBERT_{v2}$	49.9	48.5	49.2	57.0	55.1	56.1
	$BERT_{uncased}$	59.1	32.4	41.9	61.4	47.5	53.6
	$DistilBERT_{uncased}$	58.2	38.8	46.5	61.1	48.2	53.9
	ELECTRA	56.5	46.8	51.2	58.2	47.3	52.2
	RoBERTa	58.1	51.0	54.3	62.3	56.3	59.1
	XLNet	56.5	53.9	55.2	58.3	57.3	57.8
Multi	$BERT_{uncased}$	55.2	35.2	43.0	62.1	49.4	55.0
	$DistilBERT_{cased}$	55.1	45.5	49.8	57.1	54.2	55.6
	InfoXLM	57.7	54.6	**56.1**	61.2	54.5	57.6
	XLMR	58.1	48.1	52.6	62.1	52.0	56.6
	(Group 2) Cross-lingual and Multilingual Learning						
	$XLMR_{fr}$	56.9	33.2	42.0	60.0	39.1	47.3
	$XLMR_{nl}$	55.6	56.4	56.0	57.6	58.3	58.0
	$XLMR_{en,fr}$	57.2	51.2	54.0	60.4	51.5	55.6
	$XLMR_{en,nl}$	58.0	48.7	52.9	**62.4**	51.3	56.3
	$XLMR_{fr,nl}$	**60.8**	46.8	52.9	62.3	50.4	55.7
	$XLMR_{en,fr,nl}$	56.8	53.0	54.9	60.8	52.5	56.4
	$XLMR_{en,fr,nl,sl}$	45.9	66.3	54.2	48.3	65.6	55.6
	(Group 3) Label-Informed Transformers (LIT) for terminology extraction						
	LIT	37.0	70.4	48.5	45.0	65.1	53.2
	(Group 4) Mixture of Specialized Experts (MOSES) for terminology extraction						
	DeBERTa	-	-	-	59.3	56.1	57.7
	DeBERTa MoE	-	-	-	54.0	63.9	58.5
	DeBERTa MoE RNN	-	-	-	54.6	64.4	**59.1**
	(Group 5) Baselines						
	TALN-LS2N	32.6	**72.7**	45.0	34.8	**70.9**	46.7
	HAMLET	-	-	54.2	-	-	55.4
	NMT	-	-	-	-	-	55.3
	NMF	-	-	33.5	-	-	33.7

Table 4.2: The evaluation of different approaches in the French version of the ACTER corpora. The best result for each evaluation metric appears in bold for each version.

	Settings	ANN version			NES version		
		P	**R**	**F_1**	**P**	**R**	**F_1**
	(Group 1) Transformer-based models in Monolingual Learning						
Mono	CamemBERT	70.5	45.0	54.9	70.7	52.2	60.1
	FlauBERTa	**75.9**	26.2	38.9	**75.3**	39.0	51.4
Multi	$BERT_{uncased}$	67.8	37.7	48.4	69.4	49.0	57.4
	$DistilBERT_{cased}$	64.5	43.5	51.9	65.2	48.8	55.8
	InfoXLM	68.7	39.8	50.4	71.1	48.9	58.0
	XLMR	70.5	44.4	54.5	72.4	48.5	58.1
	(Group 2) Cross-lingual and Multilingual Learning						
	$XLMR_{en}$	66.7	47.9	55.8	70.6	53.8	**61.1**
	$XLMR_{nl}$	66.5	51.5	58.0	67.6	53.2	59.5
	$XLMR_{en,fr}$	63.7	52.4	57.5	68.1	52.8	59.5
	$XLMR_{en,nl}$	65.3	44.2	52.7	68.7	52.4	59.4
	$XLMR_{fr,nl}$	69.2	48.3	56.9	70.7	49.5	58.3
	$XLMR_{en,fr,nl}$	68.0	50.7	58.1	48.3	**65.6**	55.6
	$XLMR_{en,fr,nl,sl}$	58.1	**61.6**	59.8	59.5	62.5	61.0
	(Group 3) Label-Informed Transformers (LIT) for terminology extraction						
	LIT	-	-	-	-	-	-
	(Group 4) Mixture of Specialized Experts (MOSES) for terminology extraction						
	mDeBERTa	-	-	-	48.8	52.8	50.7
	mDeBERTa MoE	-	-	-	50.4	51.4	50.9
	mDeBERTa MoE RNN	-	-	-	50.0	56.4	53.0
	(Group 5) Baselines						
	TALN-LS2N	41.9	50.9	45.9	45.2	51.5	48.1
	HAMLET	-	-	**60.2**	-	-	60.8
	NMT	-	-	-	-	-	57.6
	NMF	-	-	30.9	-	-	30.7

Table 4.3: The evaluation of different approaches in the Dutch version of the ACTER corpora. The best result for each evaluation metric appears in bold for each version.

	Settings	ANN version			NES version		
		P	**R**	**F_1**	**P**	**R**	**F_1**
(Group 1) Transformer-based models in Monolingual Learning							
Mono	$BERT_{dutch\_cased}$	65.6	65.5	65.6	67.6	66.0	66.8
	robBERT	69.6	36.8	48.2	71.6	55.0	62.2
	$robBERT_{v2}$	71.6	36.4	48.3	73.6	55.7	63.4
Multi	$BERT_{uncased}$	70.7	62.5	66.3	72.3	63.7	67.8
	$DistilBERT_{cased}$	69.8	61.3	65.3	69.5	66.2	67.8
	InfoXLM	70.4	66.7	**68.5**	73.5	64.2	68.6
	XLMR	70.3	62.2	66.0	73.3	61.5	66.9
(Group 2) Cross-lingual and Multilingual Learning							
	$XLMR_{en}$	69.2	61.1	64.9	73.0	63.0	67.6
	$XLMR_{fr}$	72.1	51.0	59.8	73.6	55.5	63.3
	$XLMR_{en,fr}$	72.5	61.7	66.7	73.1	63.5	68.0
	$XLMR_{en,nl}$	69.3	60.2	64.4	74.4	61.7	67.4
	$XLMR_{fr,nl}$	**75.7**	56.7	64.8	**76.7**	59.6	67.1
	$XLMR_{en,fr,nl}$	69.9	64.3	67.0	73.7	62.9	67.9
	$XLMR_{en,fr,nl,sl}$	62.7	**75.5**	**68.5**	63.6	73.7	68.3
(Group 3) Label-Informed Transformers (LIT) for Terminology Extraction							
	LIT	49.8	73.5	59.4	50.0	**80.3**	61.6
(Group 4) Mixture of Specialized Experts (MOSES) for Terminology Extraction							
	mDeBERTa	-	-	-	68.9	61.9	65.2
	mDeBERTa MoE	-	-	-	69.1	65.0	67.0
	mDeBERTa MoE RNN	-	-	-	69.0	69.8	**69.4**
(Group 5) Baselines							
	NLPLab UQAM	18.1	19.3	18.6	18.9	18.6	18.7
	HAMLET	-	-	66.1	-	-	66.0
	NMT	-	-	-	-	-	59.6
	NMF	-	-	30.1	-	-	30.3

comes to other languages. Therefore, we decided to use a model pre-trained on multilingual languages (e.g., XLMR) to explore the diversity of different languages and domains.

Cross-lingual and Multilingual Learning vs. Baselines

We compared our experiments on cross-lingual and multilingual learning (Group 2) with monolingual learning (Group 1) and with the baselines (Group 5). The results indicated that the cross-lingual and multilingual fine-tuned models outperformed the monolingual fine-tuned models when the same models (e.g., XLMR) were used, except when it came to the precision obtained by the models of monolingual learning on the French test set. Multilingual models tended to outperform cross-lingual ones, the only exception being the cross-lingual model trained in Dutch and applied to the English test set. By adding four domains of Slovenian corpus to the training set, the multilingual model demonstrated a significant improvement in recall across all test languages, which, on average, increases by 18.2 percentage points in the ANN test set and 13.5 percentage points in the NES test set compared to monolingual learning. However, fine-tuning multilingual data could be computationally more expensive as a trade-off, especially if the dataset was large and diverse (e.g., the combination of ACTER and RSDO5 as the training set). Overall, the best token classifier for each language in this setting included [1] XLMR fine-tuned in Dutch for both versions of the English test set, [2] XLMR fine-tuned in English for the French NES version, and [3] XLMR fine-tuned in all four languages for the French ANN and both versions of the Dutch test set.

Compared to the baselines, our best cross-lingual and multilingual XLMR token classifier outperformed the winning solutions of TermEval 2020 competition (TALN-LS2N and NLPLab_UQAM) as well as the more recent research results of NMT, NMF, and HAMLET, except for the French ANN version where we had only a marginal gap (about 0.4 percentage points) with the HAMLET methods. Regarding multilingual evaluation, we showed that in contrast to the results of Lang et al. (2021), the addition of different languages generally improved the models slightly.

Note that English, French, and Dutch all belong to the Indo-European family and use the same Latin alphabet in the writing system but with different branches. English comes from the Germanic branch (closer to Dutch) and has a large vocabulary due to historical influences (French, Latin, Germanic). Dutch shares some vocabulary with English but has a more Germanic sound. French comes from the Italic/Romance branch with many words derived from Latin, which shows the greatest similarities with the Germanic branch as well. That explains why, for example, XLMR fine-tuned for Dutch still benefited both versions of the English test set in capturing the candidate terms and vice versa.

These findings confirmed our hypothesis H1.2: *"In a zero-shot cross-lingual setting, a token classifier achieves comparable results to monolingual training in a target language."* and H1.3: *"A token classifier trained on multilingual datasets and applied to a seen target language outperforms the monolingual models trained on the target language and cross-lingual models not trained on the target language."* if the languages come from the same or similar branches in the Indo-European family with a degree of closeness in the writing system.

LIT vs. Baselines

We compared the performance of the LIT architecture on terminology extraction (Group 3) in English and Dutch corpora with the baseline (Group 5) to address the impact of label information on our specific task.

Compared to the vanilla BERT architecture (pre-trained on monolingual and multilingual datasets, respectively), LIT demonstrated a significant improvement in the number of correct terms extracted via recall for both versions of the ACTER corpora but only surpassed the performance in F_1 with a 6.6 percentage point increase of the English ANN

test set. Overall, LIT indicated a discrepancy in the extraction of the ANN and NES versions where it performed better on the version without the named entities in the gold standard. We hypothesized that this behavior is due to the different lengths of terms in the two annotation categories, which is mainly due to the inclusion of long-named entities.

Compared to baselines (Group 5), LIT also outperformed the winning solution for English (TALN-LS2N) and Dutch (NLPLab_UQAM); and the NMF approaches. In particular, there was an increase of 41.3 (ANN set) and 42.9 (NES set) percentage points for LIT compared to NLPLab_UQAM in the Dutch test set. Meanwhile, our architectures delivered competitive performance with the HAMLET methods with an average difference of only 6.8 and 3.3 percentage points in F_1 for the ANN and NES versions, respectively.

We acknowledge this behavior as we have only one label type (binary labels) in terminology extraction, unlike NER tasks. As a result, the effects of informed label semantics cannot be teased out in comparison to other similar sequence labeling tasks (e.g., we captured the overall improvements of this additional information with multi-label tasks such as Historical NER).

Thus, our hypothesis H1.4 was partially confirmed:*"The integration of label semantic information into a token classifier based on BERT outperforms the base model."*. Further analysis at the term-type level should be conducted in the future to confirm this.

MOSES vs. Baselines

We reported the results of our approach on the NES version of the ACTER corpora for terminology extraction tasks and compared them on three levels: [1] vanilla (m)DeBERTa; [2] (m)DeBERTa with MoE on top; [3] (m)DeBERTa with MoE and RNN on top against the baselines (Group 5).

Compared to the vanilla architecture with a dense token classification head, the inclusion of MoE layers with and without RNN on top of the (m)DeBERTa model continuously improved performance in all languages analyzed. While there was a marginal increase when adding only the MoE layer (0.8 percentage points increase in English, 0.2 in French, and 1.8 in Dutch), stacking both the MoE and RNN layers on top of (m)DeBERTa resulted in significant performance improvements, reaching up to 4.2 percentage points increase in F_1 compared to the vanilla model and up to 2.4 percentage points increase compared to the MoE-only version.

An evaluation of our experimental results against benchmarks showed a significant improvement in F_1 and a favorable balance between precision and recall, compared to the winning solutions of TermEval 2020, NMT, and HAMLET techniques. Although our methods exhibited competitive performance in English and Dutch corpora compared to neural-supervised approaches, they achieved slightly lower scores than most neural algorithms and NMF in French. However, our approach surpassed the best neural-supervised classifier in English and Dutch, respectively.

This confirmed our hypothesis H1.5: *"A novel token classification head architecture that combines a mixture of experts (MoE) and recurrent neural networks (RNN) on a transformer-based model outperforms the base token classification model."*

4.2.3.2 RSDO5 Corpus

We reported the result of monolingual (Group 1), and multilingual (Group 2) pre-trained models in monolingual learning, and the best classifier from two groups in multilingual learning (Group 3) in comparison with the baseline (Group 4) in Table 4.4, 4.5, 4.6, and 4.7. Note that each data combination in the column name is in the form of *"[training set 1-training set 2][validation set]"* and the test set is mentioned in the table title. The best result for each evaluation metric appears in bold for each version.

Monolingual learning vs. Baselines

Table 4.4: The evaluation for the Linguistics (ling) test set in RSDO5 corpus.

Models	[bim-kem][vet]			[bim-vet][kem]			[kem-vet][bim]		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Monolingual Pre-trained Model									
SloBERTa	**73.2**	70.5	**71.8**	**73.9**	**73.5**	**73.7**	**74.5**	**74.0**	**74.2**
Multilingual Pre-trained Models									
XLMR	69.6	64.1	66.7	66.2	72.4	69.2	69.5	73.7	71.5
InfoXLM	68.4	**71.4**	69.8	67.7	71.5	69.6	73.7	66.9	70.1
$BERT_{uncased}$	66.8	65.9	66.3	66.8	68.0	67.4	66.0	69.6	67.8
$DistilBERT_{cased}$	61.8	53.4	57.3	59.1	67.2	62.9	60.9	58.2	59.5
Multilingual Learning									
$SloBERTa_{+ANN}$	67.7	69.6	68.6	66.5	71.4	68.8	69.8	66.2	67.9
$SloBERTa_{+NES}$	67.2	69.9	68.5	67.9	69.0	68.5	67.8	68.5	68.2
Baselines									
KAS-term	52.2	25.4	34.1	52.2	25.4	34.1	52.2	25.4	34.1
TermoUD	25.0	-	-	25.0	-	-	25.0	-	-

Table 4.5: The evaluation for the Biochemistry (bim) test set in RSDO5 corpus.

Models	[vet-kem][ling]			[vet-ling][kem]			[ling-kem][vet]		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Monolingual Pre-trained Model									
SloBERTa	**68.0**	**67.4**	**67.7**	**69.0**	66.6	**67.8**	**67.2**	**67.8**	**67.5**
Multilingual Pre-trained Models									
XLMR	62.3	65.2	63.7	62.4	64.0	63.2	63.5	66.8	65.1
InfoXLM	63.6	60.6	62.1	56.7	**67.5**	61.6	60.6	64.0	62.3
$BERT_{uncased}$	62.6	60.9	61.7	65.3	58.3	61.6	62.7	63.6	63.2
$DistilBERT_{cased}$	57.8	55.8	56.8	60.6	56.4	58.4	62.0	52.4	56.8
Multilingual Learning									
$SloBERTa_{+ANN}$	60.5	63.8	62.1	65.7	59.2	62.3	61.1	64.9	63.0
$SloBERTa_{+NES}$	62.6	62.3	62.4	61.8	67.1	64.3	60.9	66.7	63.7
Baselines									
KAS-term	53.8	24.8	33.9	53.8	24.8	33.9	53.8	24.8	33.9
TermoUD	21.0	-	-	21.0	-	-	21.0	-	-

We compared the performance of different monolingual (Group 1) and multilingual pre-trained models (Group 2) on the RSDO5 corpus. Both types of pre-trained models, where we used two domains from the RSDO5 corpus for training, validated on the third domain, and tested on the last domain, proved to have relatively consistent performance across all the combinations. The model performed slightly better on the *linguistics* and *veterinary* domains than on *biomechanics* and *chemistry*. Moreover, a significant performance boost was observed on the *linguistics* domain when the model was trained in the *chemistry*

Table 4.6: The evaluation for the Chemistry (kem) test set in RSDO5 corpus.

Models	**[bim-vet][ling]**			**[bim-ling][vet]**			**[ling-vet][bim]**		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Monolingual Pre-trained Model									
SloBERTa	**72.1**	**65.9**	**68.9**	70.3	**68.5**	**69.4**	**73.5**	**67.0**	**70.1**
Multilingual Pre-trained Models									
XLMR	68.7	55.1	61.2	70.2	59.2	64.3	70.1	60.3	64.8
InfoXLM	67.8	60.4	63.9	**72.0**	56.6	63.4	71.2	59.5	64.8
$BERT_{uncased}$	65.4	59.7	62.4	65.5	63.2	64.4	67.3	54.0	59.9
$DistilBERT_{cased}$	55.7	60.5	58.0	60.2	55.8	57.9	59.5	57.7	58.6
Multilingual Learning									
$SloBERTa_{+ANN}$	68.3	59.3	63.5	69.9	58.4	63.6	69.6	61.2	65.1
$SloBERTa_{+NES}$	67.5	54.6	60.4	67.9	59.2	63.3	69.3	52.7	59.9
Baselines									
KAS-term	47.8	31.4	37.8	47.8	31.4	37.8	47.8	31.4	37.8
TermoUD	24.0	-	-	24.0	-	-	24.0	-	-

Table 4.7: The evaluation for the Veterinary (vet) test set in RSDO5 corpus.

Models	**[bim-kem][ling]**			**[bim-ling][kem]**			**[ling-kem][bim]**		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Monolingual Pre-trained Model									
SloBERTa	**77.6**	66.0	**71.3**	**78.3**	65.3	**71.2**	**76.7**	64.9	**70.3**
Multilingual Pre-trained Models									
XLMR	71.1	66.7	68.8	72.7	65.6	68.9	69.3	68.1	68.7
InfoXLM	71.0	63.7	67.2	66.9	**68.9**	67.9	72.7	63.6	67.9
$BERT_{uncased}$	68.2	61.6	64.7	68.6	65.5	67.0	69.1	60.6	64.6
$DistilBERT_{cased}$	63.8	58.7	61.1	65.8	58.2	61.8	66.0	54.0	59.4
Multilingual Learning									
$SloBERTa_{+ANN}$	71.0	65.3	68.0	69.8	68.8	69.3	69.8	**68.4**	69.1
$SloBERTa_{+NES}$	69.2	**67.4**	68.3	70.5	67.8	69.1	69.3	64.7	66.9
Baselines									
KAS-term	66.9	19.3	29.9	66.9	19.3	29.9	66.9	19.3	29.9
TermoUD	21.0	-	-	21.0	-	-	21.0	-	-

and *veterinary* domains, and for the *veterinary* domain when the model was trained in *biomechanics* and *linguistics*. In these two settings, the model achieved an F_1 of more than 68%. The monolingual SloBERTa model outperformed multilingual pre-trained approaches (Group 2) in all cases by a relatively large margin in F1. This can be explained by the fact that as SloBERTa was trained on a larger Slovenian corpus (e.g., Gigafida 2.0, Kas 1.0, Janes 1.0, Slovenian parliamentary corpus siParl 2.0, and slWaC with a total subword vocabulary of 32,000 tokens) and was focused specifically on the Slovenian language in

pretraining, it has a deeper understanding of Slovenian vocabulary and syntax (language-specific knowledge) than other multilingual pre-trained models such as XLMR. This leads to higher performance also on downstream tasks in Slovene, such as terminology extraction in our case. By employing this model and looking at the best-performing train/validation combinations for each test domain, we improved the baseline in Group 4 (e.g., KAS-term) in all domains. Our results, thus, set a new benchmark for the Slovenian corpus.
Multilingual learning vs. Monolingual learning vs. Baselines

We also explored the performance of multilingual learning approaches (Group 3) on the RSDO5 test sets compared to the previous two groups and the baselines (Group 4). We trained the model using either additional ANN or NES labels from all domains of the ACTER dataset and on two domains from the RSDO5 dataset, validated on the third RSDO5 domain, and tested on the last domain. In general, our approach outperformed the approach proposed in Ljubešić et al. (2019) (KAS-term) by a large margin on all domains and according to all evaluation metrics, especially in recall. Overall, we achieved results roughly twice as high as the approach proposed by Ljubešić et al. (2019) in F_1 for all test domains in both monolingual and multilingual learning. However, the additional information on other languages (e.g., English, French, Dutch) during fine-tuning did not help the classifier to extract the candidate terms in Slovenian better than the monolingual ones (e.g., SloBERTa).

One reason for this is that Slovenian, although it has the same European root, comes from a different branch than the other three languages. More precisely, Slovene belongs to the Balto-Slavic branch. However, Slovenian terminology also contains many terms from Latin and Greek stems (e.g. *biologija* (biology) from *bio* (life) and *logos* (word, study))). This is a common feature of many European languages, including Slavic languages, but as these stems are often combined with Slovene suffixes or prefixes, the specific morphological patterns, and syntactic structures may still benefit more from the monolingual model trained on Slovenian data. We believe that however, the main reason for the higher performance of the Slovene monolingual model is that the SloBERTa model training data contained more Slovenian data and other data types, including the Corpus of academic Slovene KAS, which is the genre rich in terminology.

These findings confirmed our hypothesis H1.3: *"A token classifier trained on multilingual datasets and applied to a seen target language outperforms the monolingual models trained on the target language and cross-lingual models not trained on the target language."* if the languages are from the same or a similar branch of the Indo-European family and share the same annotation campaign (e.g. although the English and Slovenian datasets both belong to ndo-European, we collected them from different annotation campaigns with different degrees of IAA and term definitions).

4.2.3.3 Late Fusion

Inspired by the empirical studies of different monolingual and multilingual pre-trained transformers models, we reported the improvement in performance (percentage points) of the late fusion in Figure 4.8 where we ensemble two best classifiers with intersection and union in three scenarios for ACTER corpora: the [1] best monolingual and multilingual models, [2] two best monolingual models, and [3] two best multilingual models. The performance of each combination was compared with the best performance of the single model in each specific test set (in Group 1), including InfoXLM for English ANN and Dutch, RoBERTa for English NES, and CamemBERT for French.

The improvements/decline in performance over the best single model on different languages of the ACTER dataset indicated a systematic consistent characteristic: combining the candidate term sets of the two best-performing classifiers (no matter what type of clas-

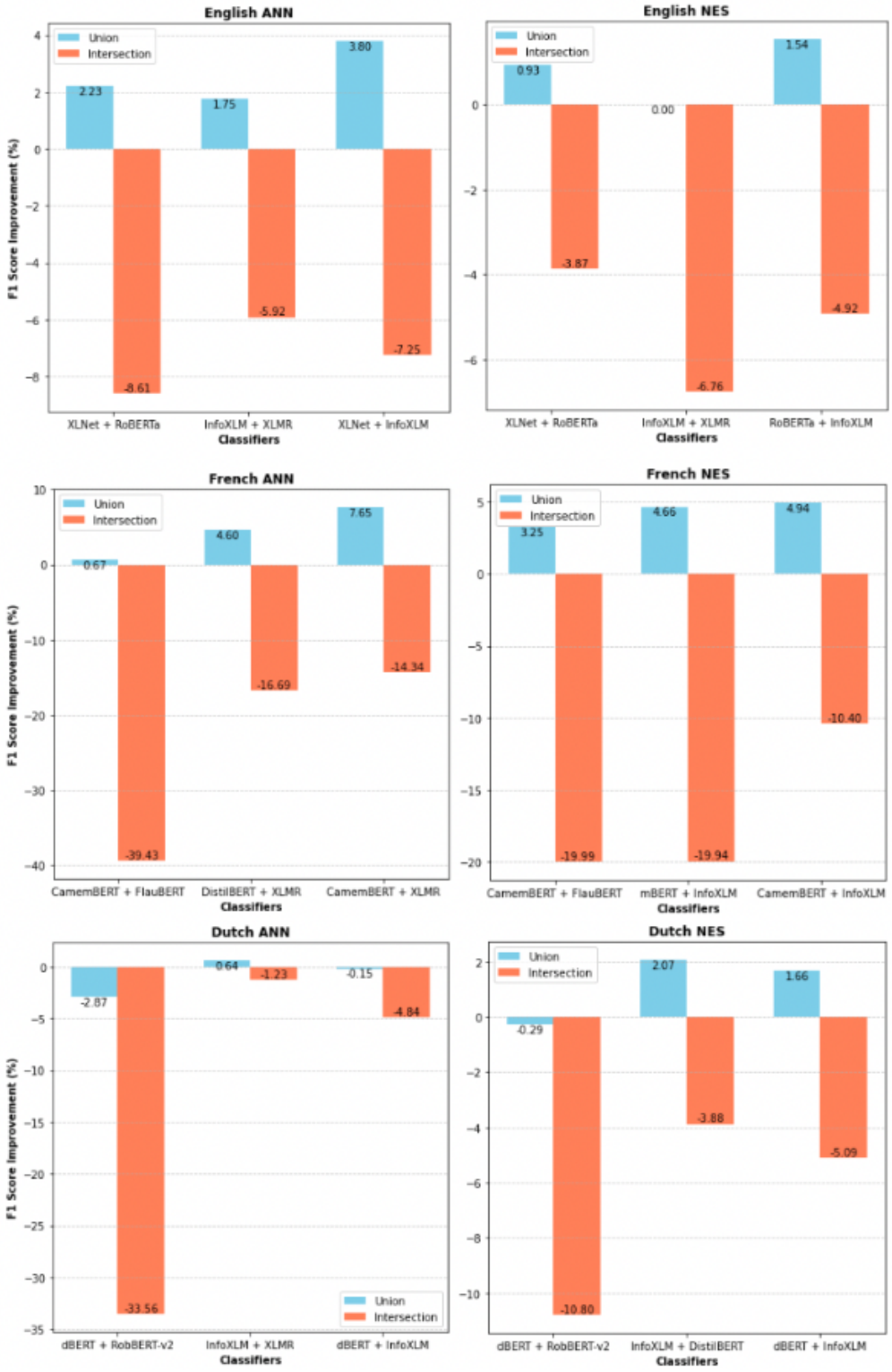


Figure 4.8: F_1 improvement using late fusion in ACTER corpora.

sifier they are) using the union always results in the biggest gain. Specifically, the union of the best monolingual and multilingual models improved the performance in most of the gold standard versions and languages except the ANN version of Dutch.

4.2.4 Error Analysis

This section focuses on understanding the predictive performance of our token classifiers through comprehensive error analysis, including [1] the impact of domain specificity in the best token classifiers for each dataset; [2] the impact of term length in the best token classifiers for each dataset; and [3] the error patterns in the prediction of the best token classifiers, which we attempt to address and mitigate in the next chapters.

4.2.4.1 The Impact of Domain Specificity

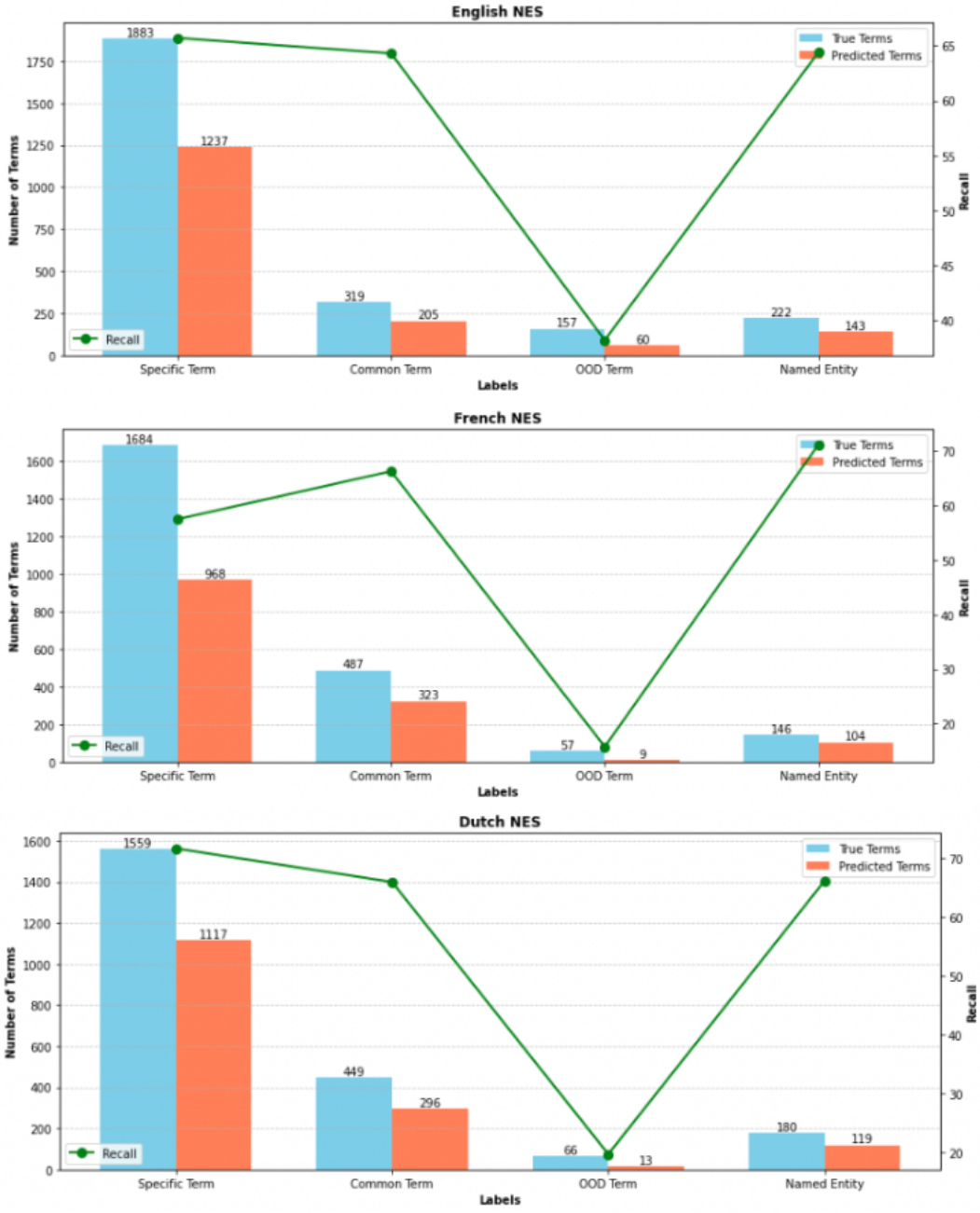


Figure 4.9: The predictive performance in recall in (green line) of (m)DeBERTa-MOE-RNN.

Although the RSDO5 corpus was limited in in-domain and out-domain types, in ACTER corpora, based on the lexicon specificity and domain specificity, Rigouts Terryn, Hoste, and Lefever (2020) introduced four labels, one for named entities and three for term labels, namely one for specific terms, one for out-of-domain (OOD) terms, and one for common terms. Regarding the impact of domain specificity and the diversity of term types, we reported the performance of different types of terms (e.g., specific terms, common terms, out-of-domain terms, and named entities) to investigate the predictive power of our best classifier on the NES version of the corpora (for the diversity of term types and the inclusion of named entities) and on each language in the corpora. We chose (m)DeBERTa-MOE-RNN as the classifier that performed the best on both the English and Dutch test sets (for consistency in behaviors).

Table 4.9 visualizes the predictive power of our classifier per type of label. The sky blue bar represents the number of true terms in the gold standard, the coral bar refers to the number of true terms that our classifier predicts as terms, and the green line represents the recall per label type. Overall, the (m)DeBERTa-MOE-RNN model could extract candidate terms belonging to specific, common terms, and named entities with competitive performance, while the model failed to retrieve OOD terms. This could be due to several factors, including [1] the rarity of OOD terms in the gold standards, and [2] the lexicon- and domain-specificity characteristics of the OOD type terms compared to other types. Regarding the earlier factor, the amount of OOD terms only covers 6.1 percent of the English gold standard and 3.0 percent of both French and Dutch ones, which made it difficult for the classifier to learn from a few examples compared to other labels. With respect to the latter factor, the OOD term type is different from the others. The specific terms are both lexicon- and domain-specific and are terms according to the strictest definitions of the concept. Meanwhile, common terms are strongly related to the domain but are not necessarily lexicon-specific. However, OOD terms are lexicon-specific, but not domain-specific. For example, in the *heart failure* corpus, some of the medical abstracts contained terminology related to statistics (e.g., "*p-value*"), which is not part of the general lexicon, but they are not very specific to the domain of *heart failure* either.

4.2.4.2 The Impact of Term Length

To determine whether the term length affects the models' performance, we calculated F_1 separately for terms of length $k = \{1, 2, 3, 4, \geq 5\}$ for both the NES version of ACTER and RSDO5 datasets. We visualized them in Figure 4.10. While the results of the ACTER dataset were obtained by employing the best-performing model (XLMR) according to the F_1 for a specific language on the *heart failure* test set, the results for the RSDO5 dataset were obtained by employing the best-performing model (SloBERTa) according to the F_1 for each domain based on each table.

The models proved to be good at predicting terms containing up to four words for English and up to three words for French and Dutch in ACTER corpora. Similarly, the results on the RSDO5 dataset showed that the models were good at predicting short terms containing up to three words for all four domains of the RSDO5 corpus. The best model applied to the *linguistics* test domain also showed relatively good performance when it came to the prediction of longer terms, achieving 45.3% for terms with at least five words. Meanwhile, in *veterinary* and *biomechanics* test domains, despite high precision for prediction of long terms, the F_1 tended to reduce due to the low recall. One factor contributing to this was the small amount of longer terms in the dataset on which the models were trained. When it came to predictions in the *chemistry* domain, there were no correct term predictions that consisted of more than five words.

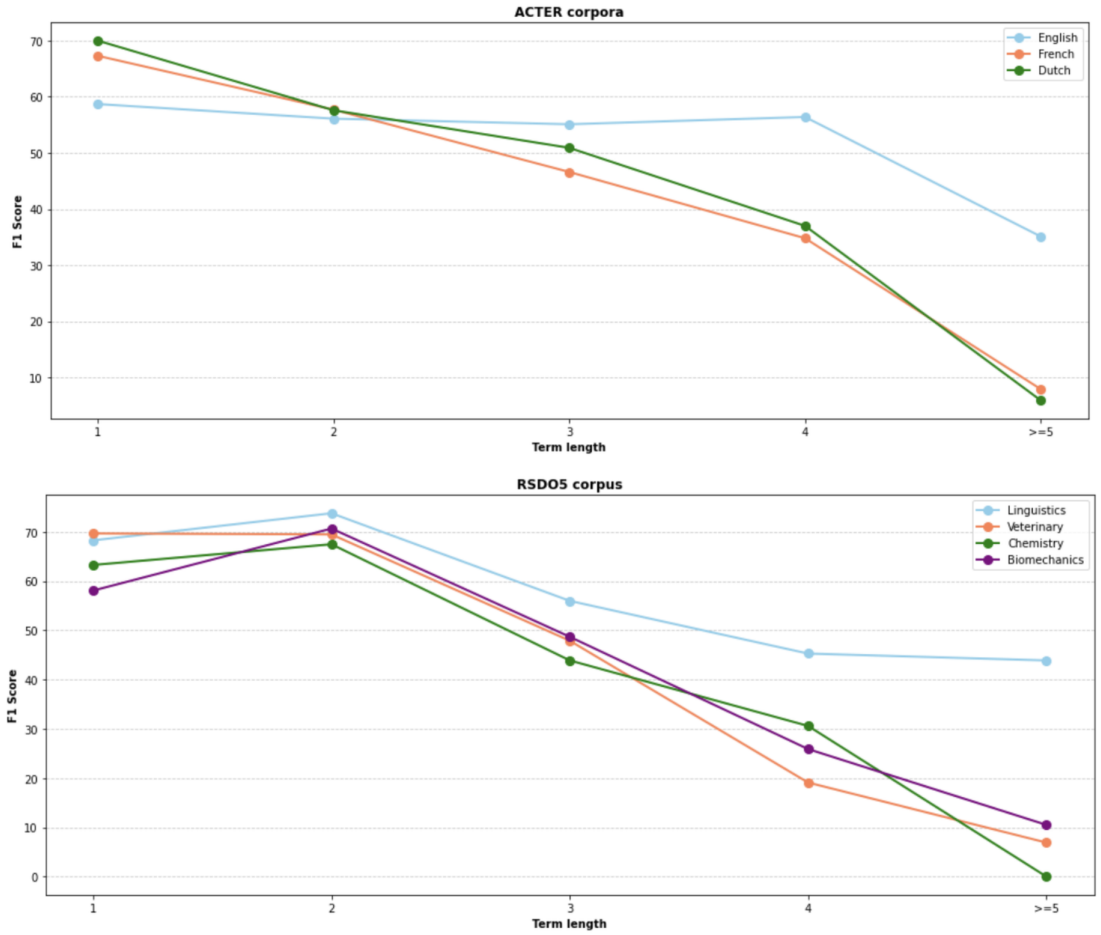


Figure 4.10: Performance in F_1 for each language in ACTER and each domain in RSDO5 sets.

4.2.4.3 The Error Patterns

In addition, we dived into different error patterns found in the list of candidate terms proposed by the model. We reported some examples of these incorrect patterns in the English version of ACTER corpora in Table 4.8 and the *linguistics* test domain in the RSDO5 corpus in Table 4.9. The first column refers to our predicted candidate term, and the second column presents the true term from the gold standard.

As the corpus contained nested terms, the very common mistake the model made was to predict a shorter term nested in the correct term of the gold standard (Pattern 1). Vice versa, the model sometimes generated incorrect predictions containing the correct nested terms (Pattern 2). Furthermore, in some cases, the model predicted a single prediction made of two consecutive terms or vice versa (Pattern 3).

The consistent error patterns in the prediction output for both datasets raised a question of whether the current BIO annotation regimes used for terminology extraction to transform the task into a sequence labeling task are not sufficient, especially for capturing nested terms. This led us to the next hypothesis, which is H2.1: *"If we apply a novel annotation regime that considers additional nested terms on terminology extraction, our token classifier can capture single nested terms better and improve the overall performance of extracting candidate terms."*

Our predictions	The gold standard
Pattern 1	
"mass spectrometric"	*"mass spectrometric analysis"*
"histologic"	*"trichrome blue histologic analysis"*
"cox proportional hazards regression"	*"multivariate cox proportional hazards regression modeling"*
...	...
Pattern 2	
"scn5a transcript"	*"scn5a"*
"cardiac rehabilitation care"	*"rehabilitation"*
"ang ii infusion mouse model"	*"infusion"*
...	...
Pattern 3	
"automatic defibrillator implantation", "resynchronization therapy"	*"multicenter automatic defibrillator implantation cardiac resynchronization therapy"*
"heart failure with reduced ejection fraction"	*"heart failure", "ejection fraction"*
"epithelial to mesenchymal transition markers"	*"epithelial", "mesenchymal"*
...	...

Table 4.8: Examples of predictions in the English heart failure domain from ACTER corpora.

Our predictions	The gold standard
Pattern 1	
"klasična analogna telefonska" (classic analog telephone)	*"klasična analogna telefonska zveza"* (classic analog telephone connection)
...	...
Pattern 2	
"brezžično slušalk v ušesu" (wireless in-ear headphones)	*"brezžično slušalk"* (wireless headphones)
"elektromehanska uporaba električne energije" (electromechanical use of electrical energy)	*"električne energije"* (electrical energy)
...	...
Pattern 3	
"batne parne stroje za pogon" (reciprocating steam engines)	*"batne parne stroje" , "pogon"* (piston steam engines), (propulsion)
"elektrarna na atomski pogon" (nuclear power plant)	*"elektrarna", "atomski pogon"* (power plant), (nuclear power plant)
"besedilnim tipom strokovnega jezika" (text type professional language)	*"besedilnim tipom", "strokovnega jezika"* (text type), (professional language)
"transformatorske postaje visoke napetosti" (high voltage transformer stations)	*"transformatorske postaje","visoke napetosti"* (transformer stations), (high voltage)
...	...

Table 4.9: Examples of predictions in the linguistics test domain from RSDO5 corpus.

4.3 Discussion

In this chapter, we investigated the effectiveness of transformer-based models as token classifiers for terminology extraction in five different directions: [1] a preliminary study to evaluate the impact of transformer-based models as token classifiers for terminology extraction on the RSDO5 corpus; [2] an empirical evaluation of different transformer-based models in monolingual learning with and without late fusion approaches; [3] multilingual and cross-lingual learning; [4] LIT: additional label information into BERT-based model; and [5] MOSES incorporating MoE with RNN layers for supervised sequence labeling tasks, including terminology extraction. While the second and third directions were tested on both ACTER and RSDO5, the last two directions were applied mainly on the ACTER sets.

Our key findings demonstrated that:

- The superiority of the multilingual pre-trained transformer-based model in recall and F_1 for terminology extracting excluding named entities in the gold standards and the improvement of ensembling different transformers models' output predictions.
- The promising impact of multilingual and cross-lingual cross-domain learning when transferring from the dominant to lesser-represented languages using the XLMR token classifier.
- The capability of our additional semantic similarity label information when compared to the conventional transformer-based architecture (e.g., BERT).
- The potential of MoE with an additional RNN layer incorporated into the conventional transformers-based architecture (e.g., DeBERTa for English, and mDeBERTa for the rest) for supervised sequence-labeling tasks, including terminology extraction with limited data.

However, we believe that there remains room for improvement in the field of supervised terminology extraction. Despite its performance in the standard BIO annotation regime, the current transformers token classifier is not optimized for nested terminology extraction. Thus, in the next chapters, we will investigate the new annotation regime that can better capture nested terms. Additionally, with the advent of large-scale language models (LLMs), we would like to test them on the ATE task and explore innovative techniques such as prompt engineering and instruction tuning to improve their performance.

Chapter 5

A Novel Nested Term Labeling Regime for ATE Tasks

As discussed previously in Chapter 4, the token classifier with the standard BIO annotation regime was unable to extract the candidate nested terms. To address this challenge and fill the gap, this chapter focuses on the impact of the annotation regime on the performance of ATE sequence-labeling models by exploring the **H2: Terminology Extraction Benefits from Nested Annotation Regime** with the following hypothesis:

- **[H2.1] The Impact of Nested Term Annotation in Terminology Extraction:** *"An annotation regime that captures additional information with regard to nested terms, improves the performance of token-based terminology extraction."*

This part resulted in the following publications:

- (H. T. H. Tran, Martinc, et al., 2024) **Hanh Thi Hong Tran**, Matej Martinc, Andraz Repar, Nikola Ljubesic, Antoine Doucet, Senja Pollak. *Can Cross-domain Term Extraction Benefit from Cross-lingual Transfer and Nested Term Labeling?*. Machine Learning, 113(7), 4285-4314. 2024.

In detail, Section 5.1 presents preliminary studies on the current status of the annotation regime on specific terminology extraction tasks and the gap with other similar downstream NLP tasks (see Section 5.1.1). Then, Section 5.1.2 introduces NOBI, the novel annotation regime specifically designed to capture the single-nested term. The results with further error analysis are discussed in Section 5.2 before the conclusion in Section 5.3.

5.1 Annotation Regimes

First, we conduct preliminary studies on different annotation regimes used for terminology extraction tasks, followed by the current regimes used for other similar downstream NLP tasks. Based on the error patterns of the best token classifier in Chapter 4, we then propose NOBI, a new annotation regime to capture nested terms, thus improving the overall performance of the terminology extraction task when the number of nested terms in the dataset is significant enough.

5.1.1 Preliminary Studies

As discussed in Chapter 1, most of the existing techniques applied to nested terminology extraction were based on either linguistic or statistical features (e.g., C-value (Š. Vintar,

2004), NPMI (Marciniak & Mykowiecka, 2015)) to rank or discard the nested terms. However, they suffered from poor results (e.g., reduced recall) due to their adaptability to new, unseen domains. Since then, no other methods have been proposed, leaving a gap in the extraction of nested terms in terminology extraction tasks.

As for other downstream NLP tasks that use the same mechanisms (e.g., NER, keyword extraction), in addition to the usual sequence-tag schemes (e.g., BIO (Ramshaw & Marcus, 1999), IOBES (Lester, 2020), BMEWO (Ratinov & Roth, 2009), BILOU (Ratinov & Roth, 2009)) for both flat and nested terms, we can categorize the methods for capturing nested entities into four main types: [1] sequence labeling, [2] hypergraph-based, [3] sequence-to-sequence (Seq2Seq), and [4] span-based methods. However, none of these methods, except for BIO and BILOU regimes for sequence-labeling tasks, has been used for terminology extraction so far.

5.1.2 NOBI Annotation Regime

Given the potential of sequence labeling adapted to the ATE task shown in Chapter 4, we consider ATE as a sequence-labeling task where the model returns a label for each token in a text sequence using two different labeling regimes: the benchmark BIO annotation scheme (Lang et al., 2021; Rigouts Terryn et al., 2021) and our novel annotation scheme called NOBI.

5.1.2.1 Description

In the BIO regime, B stands for the word with which the term begins, I stands for the word within the term, and O stands for the word that is not part of the term. The terms of a gold standard list are first mapped to the tokens in the raw text, and each word inside the text sequence is annotated with one of three labels (see the upper example in Figure 5.1). However, the BIO annotation scheme is not optimized for extracting nested terminology. Thus, we propose NOBI, an annotation regime with two additional labels, BN and IN, which refer to a word being at the beginning and within the nested term, respectively.

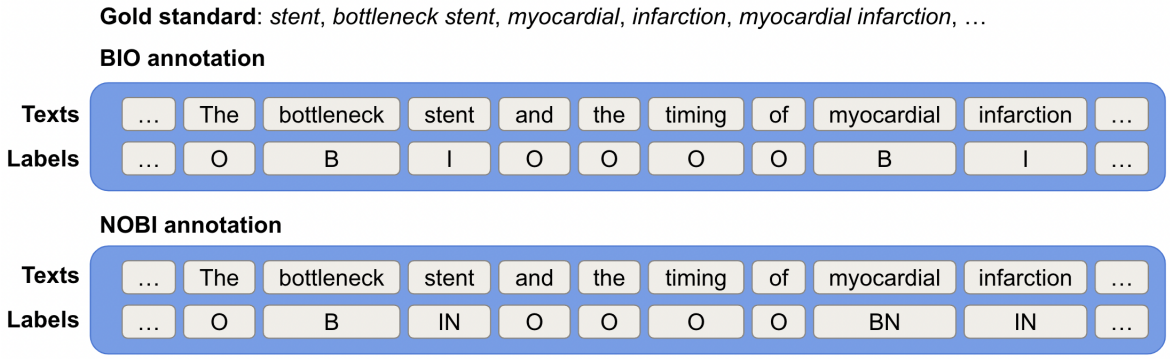


Figure 5.1: An example of BIO and NOBI annotation regimes in the ACTER corpus.

In Figure 5.1, for example, the gold standard contains the following terms: *"stent", "bottleneck stent", "myocardial", "infarction", "myocardial infarction"*, etc. In the BIO regime, we ignore the single nested terms, marking *"bottleneck"* as the beginning (B) and *"stent"* as the inside (I) of the full term *"bottleneck stent"*. Similarly, *"myocardial"* is the beginning (B), and *"infarction"* is inside (I) of the full term *"myocardial infarction"*. However, in the NOBI regime, we consider *"bottleneck stent"* and *"stent"* as two different terms, where *"stent"* is

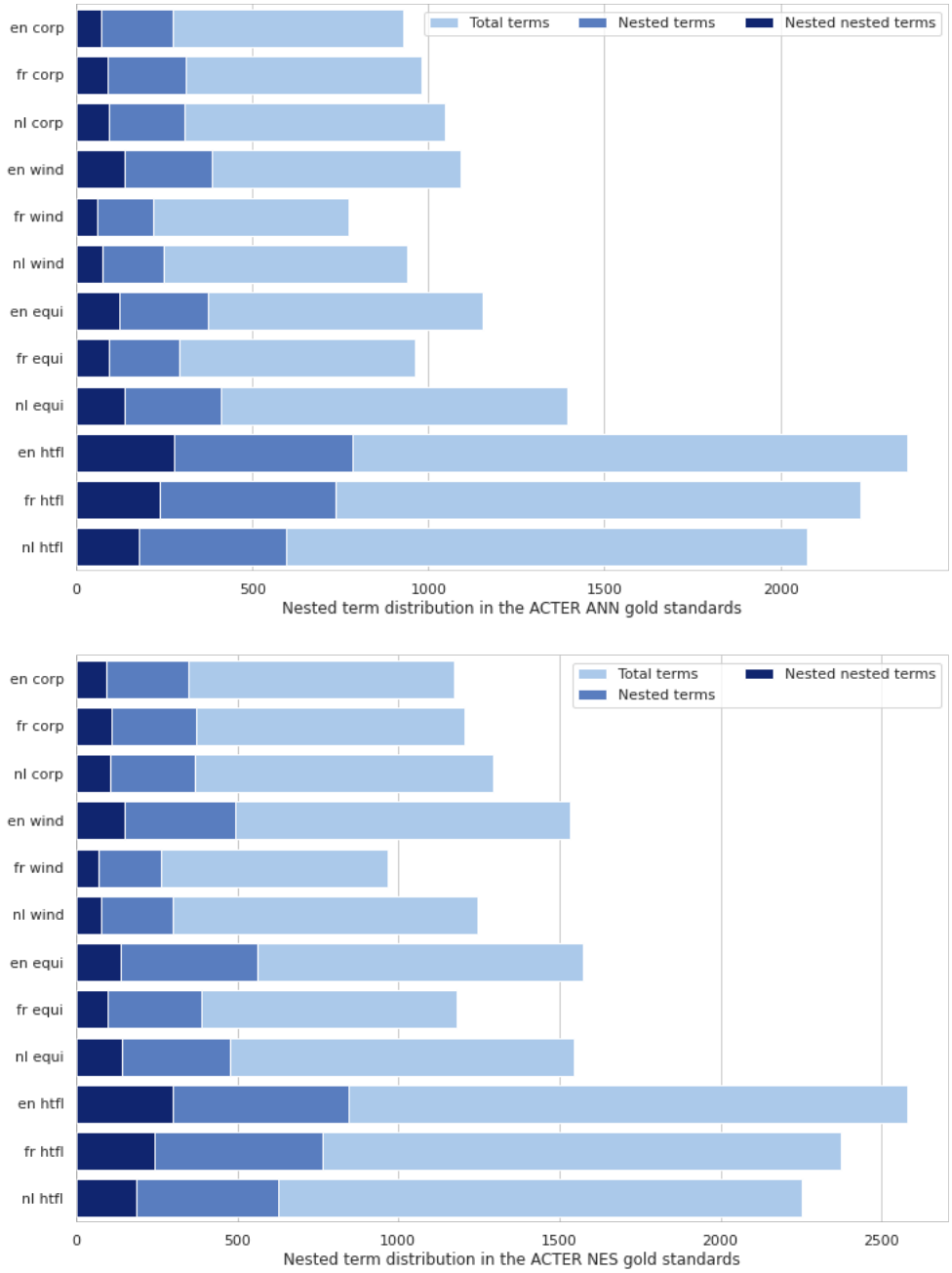


Figure 5.2: The proportion of unique nested terms in the ACTER gold standards.

the nested term of *"bottleneck stent"*. This contrasts with the BIO regime, where the model only extracts the *"bottleneck stent"* as a term. Similarly, *"myocardial"* and *"infarction"* are two separate terms nested from *"myocardial infarction"*. Therefore, an additional label N is added to the label of *"stent"*, *"myocardial"*, and *"infarction"*.

We do not consider multi-word nested terms nor terms nested within other nested terms – i.e, nested terms at the second or higher level – due to their rarity in the corpora and the gold standards (see the nested frequency in the gold standard from Figure 5.2 and

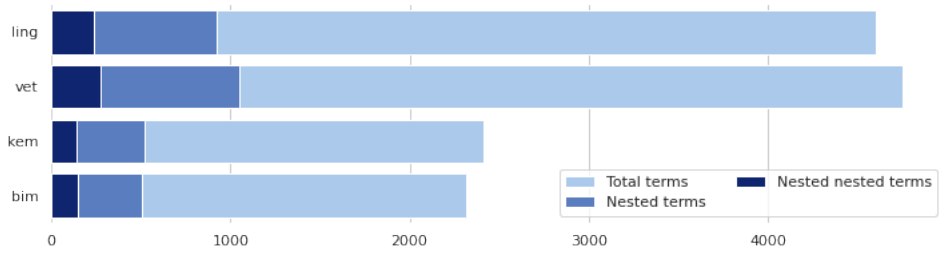


Figure 5.3: The proportion of unique nested terms in the RSDO5 gold standards.

5.3). Despite the different number of terms across languages and domains, the percentage of unique nested terms is reasonably consistent across languages and domains and is around a third of the total unique terms in the gold standards. However, the number of terms nested in other nested terms accounts for only one-tenth and one-twelfth of the total number of unique terms in both corpora, respectively, and the amounts are even much smaller if we specify the ratio per nested level (e.g., in the second level, third level).

Table 5.1: The proportion of unique nested terms of different word lengths in each domain and language of ACTER and RSDO5 corpora.

Languages	**Domains**	**k = 1**	**k = 2**	**k = 3**	**k = 4**	**k ≥ 5**	**% (k = 1)**
ACTER							
en	**corp**	246	89	11	1	1	70.69
	equi	469	87	5	1	0	77.90
	wind	282	171	36	4	0	83.51
	htfl	580	183	55	20	6	83.45
fr	**corp**	289	59	19	2	2	87.60
	equi	339	32	13	3	0	86.97
	wind	192	38	24	6	1	57.20
	htfl	620	99	30	8	9	73.56
nl	**corp**	309	46	12	2	1	84.90
	equi	414	44	12	6	0	68.72
	wind	253	36	4	4	1	80.94
	htfl	574	46	4	4	0	91.40
RSDO5							
sl	**ling**	737	177	8	0	0	79.93
	vet	835	199	13	5	1	79.30
	kem	388	126	7	2	1	74.05
	bim	349	111	17	16	14	68.84
Average							78.06

In Table 5.1, we present the proportion of nested terms with different word lengths k where $k = \{1, 2, 3, 4, \geq 5\}$ for each domain and language of both corpora. The last column on the right shows the percentage of single-word nested terms in total nested terms in the first level. On average, the proportion of single-word nested terms accounts for 78.06% of all the nested terms at the first level in the corpora.

5.1.2.2 Annotation Process

The annotation process to create the NOBI regime consists of three algorithms: [1] identifying nested terms that overlap with other terms in the ground truth (Algorithm 1); [2] creating a list of terms for each sentence or segment in the annotated data. (Algorithm 2); and [3] re-annotating the data to include NOBI annotations and ensuring that overlapping and nested boundaries are accurately captured (Algorithm 3). These algorithms provide a systematic approach to identify and re-annotate nested terms for sequence labeling tasks.

Algorithm 1 Generate List of Nested Terms

Require: DataFrame df, List of Ground Truth Terms gts
Ensure: List of Nested Terms
Initialize an empty list $nested\_terms$
for each i from 0 to $len(df) - 1$ **do**
  for each ent in $df.entities.iloc[i]$ **do**
    Extract $term$ from $df.words.iloc[i][ent[1] : ent[2] + 1]$
    Convert $term$ to lowercase
    if the number of occurrences of $term$ in $gts > 1$ **and** $term$ not in $nested\_terms$ **then**
      Append $term$ to $nested\_terms$
    end if
  end for
end for
return List of nested terms in lowercase

Algorithm 1 identifies nested terms in the data by comparing the terms extracted from the DataFrame with a list of ground truth terms. If a term occurs multiple times in the ground truth, it is considered nested. Meanwhile, Algorithm 2 extracts terms from the entities in each row of the DataFrame and stores them in a new column called *term_list*. Algorithm 3 reannotates the data by identifying and marking nested terms in the annotations. It uses the previously defined algorithms to convert the nested annotations and create the term list.

Algorithm 2 Get Term List

Require: DataFrame df
Ensure: DataFrame with Term List Column
Create a new column $term\_list$ in df
for each i from 0 to $len(df) - 1$ **do**
  Initialize an empty list li
  for each x in $df.entities.iloc[i]$ **do**
    Extract $term$ from $df.words.iloc[i][x[1] : x[2] + 1]$
    Append $term$ to li
  end for
  Assign li to $df.term\_list.iloc[i]$
end for
return DataFrame df with $term\_list$ column

Algorithm 3 Reannotate Data

Require: Input Path *input_path*, Ground Truth Path *gs_path*, Output Path *output_path*
Ensure: Ground Truth Terms and Updated DataFrame

```
Load DataFrame df from input_path
Update df using the Get Term List algorithm
Load Ground Truth Terms gts from gs_path
Identify nested_terms using the Generate List of Nested Terms algorithm
for each i from 0 to len(df) − 1 do
  for each ent in df.entities.iloc[i] do
    Extract term from df.words.iloc[i][ent[1] : ent[2] + 1]
    Initialize an empty list term_subset
    for each n from 1 to 10 do
      Generate n-grams of term and append to term_subset
    end for
    Convert term_subset to lowercase
    Identify inside_term by finding nested terms in term_subset excluding the original term
    Initialize an empty list single_inside_term
    for each x in inside_term do
      Split x into individual words and append to single_inside_term
    end for
    Remove duplicates from single_inside_term
    if len(single_inside_term) > 0 then
      for each j from ent[1] to ent[2] do
        if df.words.iloc[i][j] is in single_inside_term then
          Update df.labels.iloc[i][j] to indicate nested term (e.g., N − Term)
        end if
      end for
    end if
  end for
end for
Save updated df to output_path
return Ground Truth Terms gts and Updated DataFrame df
```

The annotations are provided in simple UTF-8 encoded plain text files. No lemmatization was performed. Aligned with the up-to-date ACTER dataset (version 1.5), the updated version with NOBI annotation has been structured as visualized in Figure 5.4.

Each annotated file (e.g., *en_corp_nes.csv*) contains the following columns: *words*, *labels*, *entities*, *term_list*. The *words* column contains the original text, the *labels* column contains the NOBI annotation, the *entities* column contains the position of the terms in the sentence, the *term_list* column contains the terms extracted from the text. For example, the first 5 rows of the *en_corp_nes.csv* file are shown in Figure 5.5.

Compared to the previous version (e.g., ACTER corpora version 1.5), we further provide the position of the terms in the sentence in the *entities* column, and the *term_list* column contains the terms extracted from each sentence in each row for further usage.

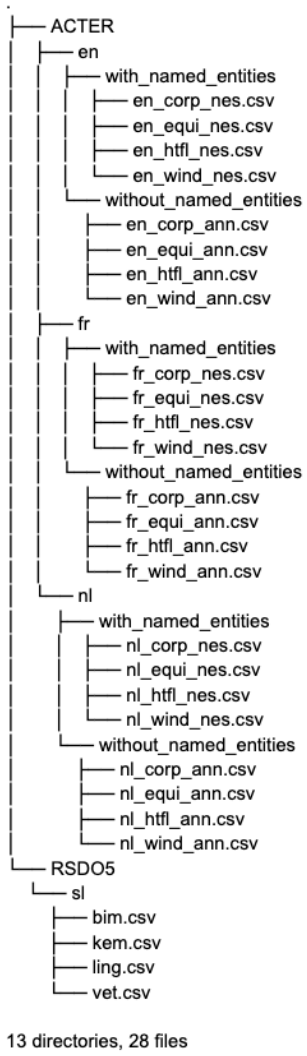


Figure 5.4: An example of *corruption* domain from ACTER corpora with NOBI annotation.

5.1.2.3 Experimental Setup

We evaluated the cross-domain performance of XLMR per annotation regime (e.g., BIO and NOBI) in monolingual, cross-lingual, and multilingual settings (as in Chapter 4).

1. **Monolingual setup.** We evaluate how well the model performs when a language-specific training corpus is available and there is a match between the language of the training set and the language of the test set. We train three monolingual models for three languages (i.e., English, French, and Dutch) and test each model in the same language for each annotation regime.

2. **Cross-lingual setup.** We evaluate the model's ability to apply the knowledge learned in one or more languages for the ATE in another, unseen language. Therefore, we fine-tune the ATE model in one or more languages (e.g., English and Dutch) and test it in another language that is not included in the training set (e.g., French). In this scenario, we examine how well the model performs without the language-specific training corpus and how well the knowledge transfer between different languages is.

3. **Multilingual setup.** We fine-tune our model by using [1] training datasets from

words	labels	entities	term_list
['Protecting', 'the', 'EU', ""'s"", 'financial', 'interests', '-', 'fight', 'against', 'fraud']	['O', 'O', 'B-Term', 'O', 'BN-Term', 'I-Term', 'O', 'B-Term', 'I-Term', 'IN-Term']	{('Term', 2, 2), ('Term', 4, 5), ('Term', 7, 9)}	['EU', 'financial interests', 'fight against fraud']
['Since', '1995', ',', 'a', 'convention', 'has', 'been', 'in', 'place', 'which', 'seeks', 'to', 'protect', ',', 'under', 'criminal', 'law', ',', 'the', 'financial', 'interests', 'of', 'the', 'EU', 'and', 'its', 'taxpayers', '.']	['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'BN-Term', 'IN-Term', 'O', 'O', 'BN-Term', 'I-Term', 'O', 'O', 'B-Term', 'O', 'O', 'B-Term', 'O']	{('Term', 26, 26), ('Term', 19, 20), ('Term', 15, 16), ('Term', 23, 23)}	['taxpayers', 'financial interests', 'criminal law', 'EU']
['Over', 'the', 'years', ',', 'the', 'Convention', 'on', 'the', 'Protection', 'of', 'the', 'European', 'Communities', ""'"", 'Financial', 'Interests', 'has', 'been', 'supplemented', 'by', 'a', 'series', 'of', 'protocols', '.']	['O', 'O', 'O', 'O', 'O', 'B-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']	{('Term', 5, 15)}	["Convention on the Protection of the European Communities ' Financial Interests"]
['ACT']	['O']	{}	[]
['Council', 'Act', 'of', '26', 'July', '1995', 'drawing', 'up', 'the', 'Convention', 'on', 'the', 'protection', 'of', 'the', 'European', 'Communities', ""'"", 'financial', 'interests']	['B-Term', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'B-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'I-Term', 'IN-Term', 'IN-Term']	{('Term', 9, 19), ('Term', 0, 0)}	["Convention on the protection of the European Communities ' financial interests", 'Council']

Figure 5.5: An example of *corruption* domain from ACTER corpora with NOBI annotation.

the languages of the ACTER dataset (i.e., English, French, and Dutch) or [2] training datasets from the languages of the ACTER dataset plus the Slovenian training dataset from the RSDO5 corpus, and then apply the model to the test datasets of all languages of the ACTER and RSDO5 dataset. In this way, we investigate whether adding more data from other languages to the training set in the target language improves the predictive performance of the model.

All three settings are applied in a cross-domain evaluation scenario, where we use two domains for training, another domain for validation, and the rest for testing. An exception is the multilingual and cross-lingual settings with the additional Slovenian corpus in the training set, where we use two domains from the ACTER corpora and all domains from the RSDO5 corpus for training. In this way, we can evaluate the generalizability of the model to adapt knowledge in one or more domains to a new unseen arbitrary domain. In the ACTER dataset, we use the *corruption* and *wind energy* domains for training, the *equitation* domain for validation and the *heart failure* domain for testing to allow direct comparison with other benchmarks that use the same train-validation-test setting (Lang et al., 2021). In the meantime, we investigate different combinations of training, validation, and testing in the RSDO5 corpus.

We divide the dataset into training, validation, and test sets. The model is fine-tuned on the training set to predict the probability that each word in a word sequence is either a part of the term (B, I), a nested term (BN for nested terms at the beginning of a multi-word term, IN for nested terms at non-beginning positions of a multi-word term) or not a part of the term (O). For this purpose, an additional token classification head is added to each model, which contains a feedforward layer with a softmax activation. The training mechanism is the same as for the token classifier (e.g. XLMR) for the dataset with the

BIO annotations, with the exception that when extracting the candidate term set, we additionally included every single word that has an N label (e.g. BN and IN) in the final list.

We evaluated the BIO and NOBI annotation systems separately using the XLMR token classifier. As explained earlier, we chose XLMR because it is competitive among other language models, as shown by our previous empirical studies for rich and lesser-known European languages. The model is first trained to predict a label for each token in the input text sequence (i.e., we model the task as a token classification problem) and then applied to the unseen text (test data). Finally, the final list of candidate terms for the test data is compiled from the tokens or token sequences labeled as terms. Note that when using the NOBI annotation regime, the terms labeled BN and IN are included separately in the final term list along with the nested terms.

Similarly to our experimental setup in Section 4, we used the XLMR token classifier as the base model and evaluated the predictive performance of each classifier when using the BIO and NOBI annotation regimes to label the training and validation sets separately. All experiments were performed on 2 GPUs of A100 with a hyperparameter configuration of 20 epochs, a learning rate of 2e-05, and a sequence length of 512.

5.2 Results

In this section, we compare the performance of the NOBI annotation regime with the BIO standard regime in the ACTER and RSDO5 corpora and report the results with general observations in Section 5.2.1. We then present a more detailed error analysis in Section 5.2.2 to gain deeper insight into the predictive behavior of the token classifier when using the NOBI regime.

5.2.1 Quantitative Results

We reported the performance of our standard token classifier when using NOBI as the annotation regime for the training corpora compared to the same token classifier when using the BIO annotation. Both systems use the same evaluation methods by comparing the aggregated list of candidate terms extracted at the level of the whole test set with the manually annotated gold standard term list using precision, recall, and F_1 (as in Chapter 4). Since the same gold standard is used to evaluate the performance of both systems, we can transparently compare their performance.

5.2.1.1 ACTER Corpora

The evaluation of our NOBI annotation regime compared to the standard BIO annotation regime with the same XLMR token classifiers in different settings (e.g., monolingual learning, cross-lingual learning, and multilingual learning) can be found in Table 5.2, 5.3 and 5.4. For each test set, the best model in precision (P), recall (R), and F_1-score (F_1) for each data version (ANN and NES) and each annotation regime separately (BIO and NOBI) is marked in bold. The arrows are used to compare BIO and NOBI for each setting, where ↑ indicates the better performance of NOBI compared to the BIO regime, while ↓ denotes the lower performance of NOBI compared to the BIO regime. In blue, we show the best model in F_1 for each test set.

Regardless of the annotation scheme, the results showed that the cross-lingual and multilingual models tended to outperform the monolingual models in all evaluation metrics in both the ANN and NES versions of the test data, except for the precision achieved by the monolingual French model on the French test set when the BIO regime was used,

Table 5.2: Evaluation on the English ACTER dataset given heart failure as a test set.

Train language	ANN						NES					
	BIO			NOBI			BIO			NOBI		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
en	58.1	48.1	52.6	↓ 57.5	↑ 48.6	↑ 52.7	62.1	52.1	56.7	↓ 58.6	↑ 55.2	↑ 56.9
fr	56.9	33.2	42.0	↓ 54.2	↑ 34.7	↑ 42.3	60.0	39.1	47.4	↓ 57.8	↑ 44.3	↑ 50.2
nl	55.6	56.4	**56.0**	↑ 57.6	↑ 58.4	↑ **58.0**	54.4	57.7	56.0	↑ 56.9	↑ 61.2	↑ 59.0
fr, sl	47.1	65.8	54.9	↓ 42.5	↑ 68.8	↓ 52.5	49.2	64.3	55.7	↓ 44.6	↑ 66.6	↓ 53.4
nl, sl	45.7	**66.3**	54.1	↑ 46.0	↑ 67.8	↑ 54.8	48.1	65.4	55.5	↑ 49.2	↑ 67.0	↑ 56.8
fr, nl	**60.8**	46.8	52.9	↓ 57.5	↓ 41.5	↓ 48.2	62.3	50.5	55.7	↓ 58.6	↑ 52.0	↑ 55.1
fr, nl, sl	50.0	62.4	55.5	↓ 48.3	↑ 67.2	↑ 56.2	52.1	63.2	57.2	↓ 49.5	↑ 65.3	↓. 56.3
en, fr	57.2	51.2	54.0	↑ **58.0**	51.2	↑ 54.4	60.4	51.5	55.6	↓ **59.5**	↑ 54.2	↑ 56.7
en, nl	58.0	48.7	52.9	↓ 54.0	↑ 56.1	↑ 55.0	**62.4**	51.4	56.4	↓ 57.4	↑ 58.6	↑ 58.0
en, sl	48.1	63.2	54.6	↑ 49.0	↑ 65.7	↑ 56.1	54.9	63.8	59.0	↓ 50.8	↑ 64.4	↓ 56.8
en, fr, sl	48.1	64.2	55.0	↑ 51.1	↑ 67.2	↑ **58.0**	58.4	61.1	**59.7**	↓ 55.2	↑ 63.4	↓ 59.0
en, nl, sl	48.4	65.0	55.4	↓ 44.8	↑ 68.6	↓ 54.2	54.5	63.3	58.6	↓ 53.1	↑ 67.3	↑ **59.3**
en, fr, nl	56.8	53.0	54.9	↓ 55.7	↓ 51.0	↓ 53.3	60.8	52.6	56.4	↓ 57.4	↑ 59.8	↑ 58.6
en, fr, nl, sl	45.9	**66.3**	54.2	↓ 45.5	↑ **69.3**	↑ 54.9	48.3	**65.7**	55.6	↑ 51.9	↑ **68.4**	↑ 59.0
cross-ling. avg.	52.7	55.2	52.6	↓ 51.0	↑ 56.4	↓ 52.0	54.4	56.7	54.6	↓ 52.8	↑ 59.4	↑ 55.1
multi-ling. avg.	51.8	58.8	54.4	↓ 51.2	↑ 61.3	↑ 55.1	57.1	58.5	57.3	↓ 55.0	↑ 62.3	↑ 58.2

Table 5.3: Evaluation on the French ACTER dataset given heart failure as a test set.

Train language	ANN						NES					
	BIO			NOBI			BIO			NOBI		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
fr	**70.5**	44.4	54.5	↓ 66.3	↑ 48.9	↑ 56.3	**72.4**	48.5	58.1	↓ 65.9	↓ 54.7	↑ 59.8
en	66.7	47.9	55.8	↑ 67.8	↓ 44.8	↓ 53.9	70.6	53.8	61.1	↓ 65.3	↓ 53.3	↓ 58.7
nl	66.5	51.5	58.0	↓ 64.7	↓ 47.0	↓ 55.1	67.6	53.2	59.5	↓ 67.5	↓ 53.1	↓ 59.4
en, sl	60.2	61.4	60.8	↑ 61.1	↓ 57.5	↓ 59.2	57.8	**62.5**	60.1	↑ 62.9	↓ 56.0	↓ 59.2
nl, sl	61.4	60.4	60.9	↓ 59.5	↓ 58.5	↓ 59.0	61.8	59.9	60.8	↑ 63.1	↓ 56.7	↓ 59.7
en, nl	65.3	44.2	52.7	↓ 65.2	↑ 47.9	↑ 55.2	68.7	52.4	59.4	↑ **69.3**	↓ 50.6	↓ 58.5
en, nl, sl	58.7	61.0	59.8	↓ 55.3	↑ 63.2	↓ 59.0	60.9	62.0	61.5	↓ 59.0	↑ 62.3	↓ 60.6
fr, en	63.7	52.4	57.5	↑ 65.7	↓ 49.5	↓ 56.4	68.1	52.8	59.5	↓ 67.2	↓ 49.6	↓ 57.1
fr, nl	69.2	48.3	56.9	↓ **66.4**	↑ 48.4	↓ 56.0	70.7	49.5	58.3	↓ 66.1	↑ 54.2	↑ 59.6
fr, sl	65.0	56.6	60.5	↓ 58.8	↑ 62.3	60.5	65.3	57.6	61.2	↓ 56.9	↑ 64.0	↓ 60.2
fr, en, sl	61.5	58.6	60.0	↑ 63.2	↑ 60.5	↑ **61.8**	67.4	57.5	**62.1**	↓ 64.1	↑ 61.6	↑ **62.9**
fr, nl, sl	64.9	58.2	**61.4**	↓ 61.5	↑ 61.0	↓ 61.3	65.3	57.9	61.4	↓ 63.1	↑ **62.7**	↑ **62.9**
fr, en, nl	68.0	50.7	58.1	↓ 65.4	↓ 46.9	↓ 54.6	70.2	52.1	59.8	↓ 63.8	↑ 56.5	↑ 60.0
en, fr, nl, sl	58.1	**61.6**	59.8	↑ 60.3	↑ **62.8**	↑ 61.6	59.5	**62.5**	61.0	↑ 64.2	↓ 59.5	↑ 61.7
cross-ling. avg.	63.1	54.4	58.0	↓ 62.3	↓ 53.3	↓ 56.9	64.6	57.3	60.4	↓ 64.5	↓ 55.3	↓ 59.4
multi-ling. avg.	64.3	55.2	59.2	↓ 63.0	↑ 55.9	↓ 58.9	66.6	55.7	60.5	↓ 63.6	↑ 58.3	↑ 60.6

and the monolingual Dutch model on the Dutch test set when the NOBI scheme was used. The multilingual models generally performed better than the cross-lingual models in F_1. However, the multilingual models had lower precision than the monolingual and cross-lingual models. By including the Slovenian corpus with four different domains in the training set, the multilingual model showed a significant improvement in recall in all test languages compared to the monolingual setting. It also outperformed the other models in F_1 when we evaluated it in all three test sets with both annotations. However, this improvement came at the expense of precision.

The choice of domains for training a terminology extraction model is undoubtedly crucial for its performance. Based on the above performance, we have learned some valuable guidelines for selecting domains for terminology extraction. First, there should be a certain

Table 5.4: Evaluation on the Dutch ACTER dataset given heart failure as a test set.

Train language	ANN						NES					
	BIO			NOBI			BIO			NOBI		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
nl	70.3	62.2	66.0	↑ 71.2	↑ 64.1	↑ 67.5	73.3	61.5	66.9	↑ **73.5**	↑ 62.6	↑ 67.6
en	69.2	61.1	64.9	↑ 71.0	61.1	↑ 65.7	73.0	63.0	67.6	↓ 69.4	↑ 68.4	↑ 68.9
fr	72.1	51.0	59.8	↓ 70.4	↑ 55.6	↑ 62.2	73.6	55.5	63.3	↓ 70.4	↑ 62.4	↑ 66.2
en, sl	59.5	76.6	67.0	↑ 61.6	↑ 78.3	↑ 68.9	61.1	73.6	66.7	↑ 61.6	↑ **75.7**	↑ 68.4
fr, sl	62.5	74.7	68.1	↓ 58.7	↑ **79.3**	↓ 67.5	61.6	71.2	66.1	↓ 59.4	↑ 75.1	↑ 66.3
en, fr	72.5	61.7	66.7	↓ 70.8	↓ 60.1	↓ 65.0	73.1	63.5	68.0	↓ 72.5	↓ 61.2	↓ 66.4
en, fr, sl	59.6	**77.0**	67.2	↑ 61.1	↑ 78.2	↑ 68.6	66.6	69.6	68.1	↓ 66.4	↑ 74.9	↑ 70.4
nl, en	69.3	60.2	64.4	↓ 68.6	↑ 62.7	↑ 65.5	74.4	61.7	67.4	↓ 70.7	↑ 66.3	↑ 68.4
nl, fr	**75.7**	56.7	64.8	↓ **73.2**	↑ 58.1	64.8	**76.7**	59.6	67.1	↓ 73.0	↑ 60.6	↓ 66.2
nl, sl	65.8	72.7	**69.1**	↓ 65.0	↑ 77.0	↑ 70.5	69.9	69.7	**69.8**	↓ 68.6	↑ 72.5	↑ 70.5
nl, en, sl	64.7	73.0	68.6	↓ 60.0	↑ 80.6	↑ 68.8	68.7	70.3	69.5	↓ 67.6	↑ 74.2	↑ 70.8
nl, fr, sl	69.2	69.0	**69.1**	↓ 65.2	↑ 76.5	↑ 70.4	69.4	69.4	69.4	↓ 65.4	↑ 74.4	↑ 69.6
nl, en, fr	69.9	64.3	67.0	↑ 72.1	↓ 55.5	↓ 62.7	73.7	62.9	67.9	↓ 71.1	↑ 64.9	↓ 67.8
en, fr, nl, sl	62.7	75.5	68.5	↑ 64.5	↑ 78.1	↑ **70.6**	63.6	**73.7**	68.3	↑ 69.2	↓ 73.2	↑ **71.1**
cross-ling. avg.	65.9	67.0	65.6	↓ 65.6	↑ 68.8	↑ 66.3	68.2	66.1	66.6	↓ 66.6	↑ 69.6	↑ 67.8
multi-ling. avg.	68.2	67.3	67.4	↓ 66.9	↑ 69.8	↑ 67.6	70.9	66.8	68.5	↓ 69.4	↑ 69.4	↑ 69.2

degree of relevance from the training domain to the target test set. For example, in our case, we included the Slovenian corpus in the training set of the token classifier to make predictions for the new unseen test set (e.g., *heart failure*). Since RSDO5 covers some domains (e.g. *biomechanics*, *chemistry*) that are directly relevant to the domain that the classifier is supposed to predict. Furthermore, the diversity of data sources (e.g. academic papers, news articles) and domain specificity support the performance of the classifier.

These results were consistent with our discussion in Section 4.2.3 and thus confirmed hypothesis H1.2: *"In a zero-shot cross-lingual setting, a token classifier achieves comparable results to monolingual training in a target language."* and H1.3: *"A token classifier trained on multilingual datasets and applied to a seen target language outperforms the monolingual models trained on the target language and cross-lingual models not trained on the target language."* if the languages are from the same or a similar branch of the Indo-European family and share the same annotation campaign (e.g. although the English and Slovenian datasets both belong to ndo-European, we collected them from different annotation campaigns with different degrees of IAA and term definitions).

When comparing the two annotation regimes, the use of NOBI annotations improved the recall of the model in many cases. This is especially true for monolingual and multilingual settings (see Figure 5.6, 5.7, and 5.8) - in which the models were trained in multiple languages, including the language of the test sets for all scenarios - and for cross-lingual settings - in which the models were trained in only one language and applied to the other languages except for the French test set. A significant increase in recall also improved F_1 overall.

In Table 5.5, we reported the best-performing models from our combinations of training data: [1] for the English and French test sets, the best results were obtained with English, French, and Slovenian training data; and [2] for the Dutch test set, the best results were obtained with the multilingual classifiers of all four languages. Thus, we compared the multilingual XLMR classifier fine-tuned to the pre-defined test language with the multilingual XLMR classifier (trained in at least three languages, including Slovenian and the language of the test set) using the ACTER dataset in both annotation regimes. This demonstrated the strength of a multilingual pre-trained language model with multilingual settings - using either [1] English, French, and Slovenian or [2] all four languages as a

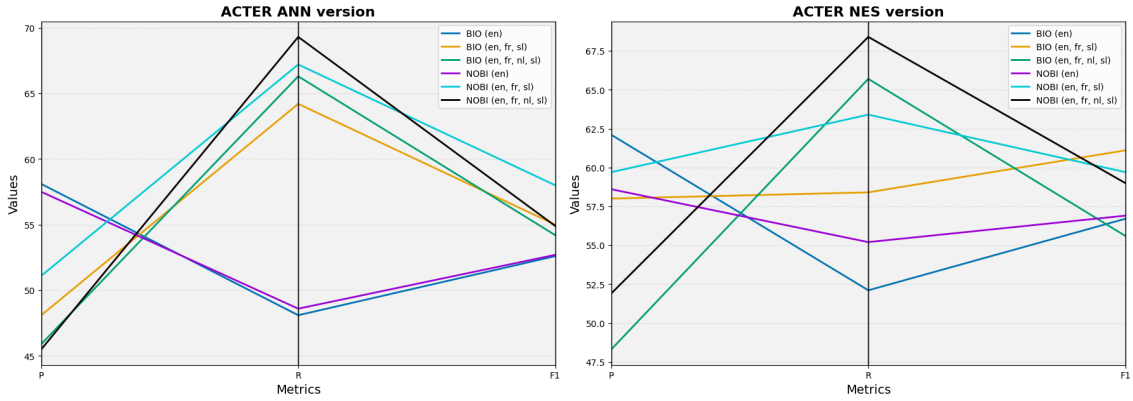


Figure 5.6: Parallel Coordinates Plot in performance of XLMR classifier for the English test set.

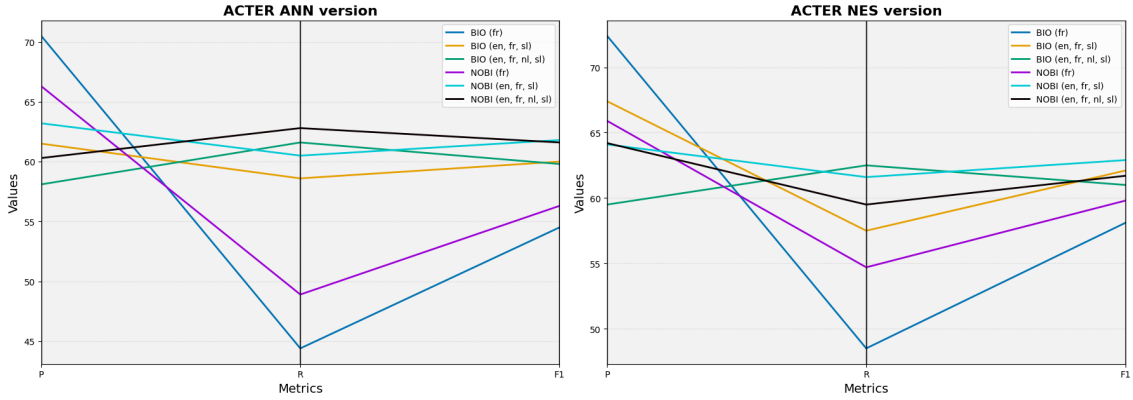


Figure 5.7: Parallel Coordinates Plot in performance of XLMR classifier for the French test set.

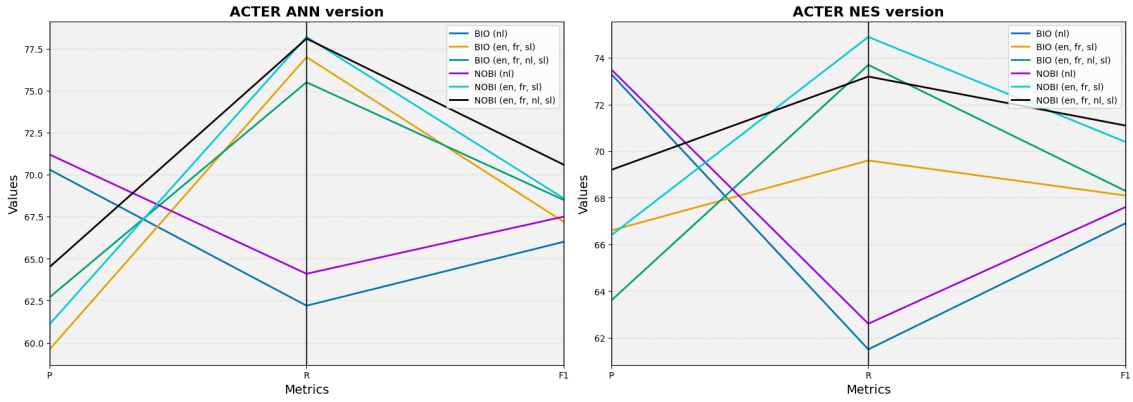


Figure 5.8: Parallel Coordinates Plot in performance of XLMR classifier for the Dutch test set.

training set - in capturing and understanding different linguistic nuances compared to a monolingual model. In addition, the NOBI regime outperformed the BIO regime in most testing scenarios.

Furthermore, in F_1, we compared the proposed results with the benchmarks shown in Table 5.5 to prove hypothesis H2.1. For comparison, we used the solutions of the winning teams of the TermEval 2020 competition (TALN-LS2N (Hazem et al., 2020) won on the

Table 5.5: F_1 comparison between our classifier and the baselines in ACTER corpora.

Methods	English		French		Dutch	
	ANN	NES	ANN	NES	ANN	NES
Baselines						
Winning teams	45.0	46.7	45.9	48.2	18.6	18.7
HAMLET	54.2	55.4	60.2	60.8	66.1	66.0
NMT	-	55.3	-	57.6	-	59.6
NMF	33.5	33.7	30.9	30.7	30.1	30.3
Our token classifier with two annotation regimes						
BIO classifier	54.9	**59.7**	61.4	62.1	69.1	69.8
NOBI classifier	**58.0**	59.3	**61.8**	**62.9**	**70.6**	**71.1**

English and French test set, while NLPLab_UQAM (N. T. Le & Sadat, 2021) won on the Dutch test set) and other recent methods such as HAMLET (Rigouts Terryn et al., 2021), NMT (Lang et al., 2021) and NMF (Nugumanova et al., 2022).

Our classifiers trained with either BIO or NOBI annotations outperformed the previously described benchmark approaches and presented significant performance improvements as measured by F_1. When comparing classifiers trained with BIO and NOBI annotation schemes, the BIO classifiers showed a better F_1 for the English NES gold standard with named entities. Meanwhile, the classifiers trained with the NOBI system demonstrated remarkable performance and outperformed all existing SOTA models, including our BIO classifiers, in all languages included in both the ANN and NES versions, except for the English NES corpus mentioned above. This confirmed our hypothesis H2.1: *"An annotation regime that captures additional information with regard to nested terms, improves the performance of token-based terminology extraction."*

5.2.1.2 RSDO5 Corpus

In addition, we applied monolingual and multilingual cross-domain approaches to the Slovenian RSDO5 dataset. The results, grouped by test domain using the BIO and NOBI annotation regimes, are presented in Tables 5.6 and 5.7, respectively. For each annotation regime, we evaluated monolingual and multilingual settings where ANN and NES versions are added to the training set of the RSDO5 corpus.

The monolingual approach, in which we used two domains from the RSDO5 corpus for training, validated the third domain, and tested the last domain, proved to be relatively consistent across all combinations in both annotation regimes. For both regimes, we achieved a precision of more than 61%, a recall of no less than 55%, and an F_1 of more than 57%. In addition, they performed slightly better in *linguistics* and *veterinary* than in *biomechanics*. The difference in the number and length of terms per domain mentioned in Section 5.2.2 could be one of the factors contributing to this behavior. In addition, a significant increase in performance was observed for the *veterinary* domain when the model was trained in the *biomechanics* and *linguistics* domains, and for the *linguistics* domain when the *veterinary* domain was included in the training set for the model in both annotation regimes. Between these two settings, the BIO regime classifier achieved a performance of up to 68.9% in F_1 for the *linguistics* test set, outperforming other domains in the same regimes and also all cases in the monolingual classifier of the NOBI regime.

We also investigated the performance of multilingual approaches in the RSDO5 test sets. We trained the model with the ANN and NES labels from all domains of the AC-

Table 5.6: The evaluation in RSDO5 corpus given each domain as a test set in a monolingual setting. Bold indicates the best result for each test set. The comparison between BIO and NOBI as well as the best model in F_1 are set in the same mechanism with Table 5.2, 5.3, and 5.4.

Valid set	Test set	BIO P	BIO R	BIO F_1	NOBI P	NOBI R	NOBI F_1
Monolingual learning							
vet	ling	**69.6**	64.1	66.7	↓ 65.4	↑65.4	↓ 65.4
bim	ling	69.5	**73.7**	**71.5**	↓ **66.9**	↓ 69.5	↓ 68.2
kem	ling	66.2	72.4	69.2	↓ 64.9	↓ **72.3**	↓ **68.4**
ling	vet	71.1	66.7	68.8	↓ 66.6	↑68.5	↓ 67.5
kem	vet	**72.7**	65.6	**68.9**	↓ 66.9	↑ **69.7**	↓ **68.3**
bim	vet	69.3	**68.1**	68.7	↓ **67.6**	↓ 62.5	↓ 65.0
ling	kem	68.7	55.1	61.2	↓ 63.8	↑ **61.4**	↑ 62.6
bim	kem	**70.2**	**60.3**	**64.8**	↓ 66.1	↑ **61.4**	↓ 63.7
vet	kem	**70.2**	59.2	64.3	↓ **68.3**	↑ 60.6	↓ **64.2**
vet	bim	**63.5**	**66.8**	**65.1**	↓ **61.4**	↓ 61.3	↓ **61.3**
ling	bim	62.3	65.2	63.7	↓ 57.2	↓ 60.1	↓ 58.6
kem	bim	62.4	64.0	63.2	↓ 61.0	↓ **61.7**	↓ **61.3**
Avg.		68.0	65.1	66.3	↓ 64.7	↓ 64.5	↓ 64.5

Table 5.7: The evaluation in RSDO5 corpus given each domain as a test set in the multilingual setting. In this setting, in addition to Slovenian training data, the data from ACTER in en, fr, and nl is used, and ANN and NES training sets are compared.

		ANN						NES					
Valid.	Test	BIO			NOBI			BIO			NOBI		
set	set	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Multilingual learning													
vet	ling	67.7	69.6	68.6	↓ **67.5**	↓ 62.7	↓ 65.0	67.2	**69.9**	**68.5**	↓ 64.2	↓ **67.3**	↓ **65.7**
bim	ling	**69.8**	66.2	67.9	↓ 64.6	↓ 68.1	↑ **66.3**	67.8	68.5	68.2	↓ **64.9**	↓ 64.8	↓ 64.8
kem	ling	66.5	**71.4**	**68.8**	↓ 59.6	↓ **71.0**	↓ 64.8	**67.9**	69.0	**68.5**	↓ 59.9	↓ 65.1	↓ 62.4
ling	vet	**71.0**	65.3	68.0	↓ 62.4	↑ **70.9**	↓ 66.4	69.2	67.4	68.3	↓ 61.8	↑ 70.8	↓ 66.0
kem	vet	69.8	**68.8**	**69.3**	↓ 68.0	↓ 68.5	↓ **68.2**	**70.5**	**67.8**	**69.1**	↓ **64.6**	↑ 70.6	↓ **67.5**
bim	vet	69.8	68.4	69.1	↓ **68.7**	↓ 67.1	↓ 67.9	69.3	64.7	66.9	↓ 63.0	↑ **72.8**	↓ **67.5**
ling	kem	68.3	59.3	63.5	↓ 66.0	↓ 52.9	↓ 58.7	67.5	54.6	60.4	↓ 62.8	↑ **60.8**	↑ 61.8
bim	kem	69.6	**61.2**	**65.1**	↓ **66.6**	↓ 55.5	↓ 60.5	**69.3**	52.7	59.9	↓ **65.5**	↑ **60.8**	↑ **63.1**
vet	kem	**69.9**	58.4	63.6	↓ 65.9	↓ **57.7**	↓ **61.5**	67.9	**59.2**	**63.3**	↓ 62.8	↑ **60.8**	↓ 61.8
vet	bim	61.2	**64.9**	**63.0**	↑ **62.9**	↓ 62.6	↓ 62.7	60.9	66.7	63.7	↓ 59.1	↓ 64.0	↓ 61.5
ling	bim	60.5	63.8	62.1	↓ 56.2	↓ 58.2	↓ 57.2	62.6	62.3	62.4	↓ 57.0	↑ 62.9	↓ 59.8
kem	bim	**65.7**	59.2	62.3	↓ 59.5	↑ **66.7**	↑ **62.9**	61.8	**67.1**	**64.3**	↓ 61.0	**67.1**	↓ **63.9**
Avg.		67.5	64.7	65.9	↓ 64.0	↓ 63.5	↓ 63.5	66.8	64.2	65.3	↓ 62.2	↑ 65.6	↓ 63.8

TER dataset and on two domains of the RSDO5 dataset, validated on the third RSDO5 domain, and tested on the last domain. Tables 5.6 and 5.7 showed the comparative performance of the multilingual and monolingual approaches. However, the results demonstrated a discrepancy in performance-enhancing efficiency between the different combinations of training, validation, and test sets. This raises the hypothesis of domain sensitivity in transfer learning for ATE tasks. Therefore, a careful selection of domains in the training set is necessary to increase the performance of the classifier (as discussed in Chapter 4.2.3).

We also compared two different annotation regimes by evaluating the performance of the classifiers with different combinations of training, validation, and testing for each system. Despite the consistency of the predictive power of the monolingual and multilingual settings, the NOBI annotation classifiers showed poorer performance than the BIO regime in the Slovenian RSDO5 corpus. This is because the proportion of nested terms in RSDO5 is too low for the classifier to learn the nested terms correctly. In Figure 5.3, the nested terms are visualized in relation to the unique nested terms and the terms nested in other nested terms. This confirmed our hypothesis H2.1: *"An annotation regime that captures additional information with regard to nested terms, improves the performance of token-based terminology extraction."* given the condition that the number of nested terms is significant in the corpus.

In Table 5.8, we presented the comparison between related works that used the RSDO5 dataset and our proposed results using monolingual and multilingual approaches. The result of the method of Ljubešić et al. (2019), which was re-implemented using the same RSDO5 corpus[1] as our studies, is from H. Tran et al. (2022). In general, our approach outperformed Ljubešić et al. (2019) by far in all domains and all evaluation metrics, especially recall. We achieved about twice as high results as the Ljubešić et al. (2019) approach in F_1 for all test domains in both monolingual and multilingual learning. It should be noted that the Ljubešić et al. (2019) method was primarily intended for extracting terms from PhD theses, i.e., from documents that are much longer than those available in our training data, which explains the low recall of this approach. However, this result highlighted a key strength of the sequence-labeling approach. Namely, it does not rely on the frequency of occurrence of terms, which makes the approach more robust, as this comparison shows. In our case, we have shown that multilingual experiments improved our monolingual results in some cases (H. Tran et al., 2022), although not systematically.

Table 5.8: Comparison between our performance and SOTA in RSDO5 dataset.

Methods	**Linguistics**			**Veterinary**			**Chemistry**			**Biomechanics**		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Baselines												
KAS-term	52.2	25.4	34.1	66.9	19.3	29.9	47.8	31.4	37.8	53.8	24.8	33.9
TermoUD	25.0	25.0	25.0	21.0	21.0	21.0	24.0	24.0	24.0	21.0	21.0	21.0
Token classifier with BIO/NOBI annotation regime												
Mono BIO	**69.5**	**73.7**	**71.5**	**72.7**	65.6	68.9	**70.1**	60.3	64.8	**63.5**	66.8	**65.1**
Multi BIO	66.5	71.5	68.8	69.8	**68.8**	**69.3**	69.6	**61.2**	**65.1**	61.8	**67.1**	64.3
Mono NOBI	64.9	72.3	68.4	66.9	69.7	68.3	68.3	60.6	64.2	61.4	61.3	61.3
Multi NOBI	64.6	68.1	66.3	68.0	68.5	68.2	65.5	60.8	63.1	61.0	67.1	63.9

5.2.1.3 Comparison on Annotation Regimes

In addition to the popular BIO regime, IOBES and BILOU are two other annotation regimes that are frequently used in NLP tasks. These regimes are used to represent and label entities within a sequence of words or tokens in a text. IOBES stands for token [**I**]nside an entity; [**O**]utside an entity (i.e., not part of an entity); [**B**]eginning token of an entity; [**E**]nd token of an entity; [**S**]ingle token that forms a whole entity by itself. The IOBES schema is an extension of the BIO regime with the additional tags "E" and "S" to

[1]This implementation was performed by the co-author A. Repar.

represent entities that end at a token or consist of a single token. BILOU, on the other hand, represents tokens [**B**]eginning token of an entity; [**I**]nside an entity; [**L**]ast token of an entity; [**O**]utside an entity; and [**U**]nit token, which in itself forms a whole entity. The BILOU regime is an extension of the IOBES regime, but provides a more compact representation of entities consisting of multiple tokens and uses the same "B", "I" and "O". We demonstrated the performance of our XLMR classifier on ACTER English sets in BIO, NOBI, BIOES, and BILOU using the ANN gold standard as shown in Table 5.9.

Table 5.9: Evaluation of XLMR fine-tuned on ACTER English sets with NES gold standard using different annotation regimes.

Models	P	R	F_1
BIO	62.1	52.1	56.7
BIOES	**62.6**	51.9	56.7
BILOU	61.8	52.6	56.8
NOBI	58.6	**55.2**	**56.9**

The results demonstrated a comparable performance of the different annotation methods. NOBI offered significant improvements in recall and thus a better balance between precision and recall. Both IOBES and BILOU were often used to label entities and were converted to simpler BIO formats during training or evaluation. These annotation regimes helped models understand the boundaries and types of entity in a text so that they learned to recognize and extract them effectively. However, the standard annotation regimes IOBES and BILOU did not support nested entities well but have been used as a basis (similar to BIO) to improve nested terms. Both IOBES and BILOU were designed to represent terms or entities in a flat manner, where each token in the text is associated with only one entity tag. In a nested entity scenario, we proposed hierarchically structured entities where an entity/term is wholly or partially contained within another entity/term. To represent nested entities, we introduced additional custom annotation regimes, namely NOBI, and a simple regime to appropriately handle the single-word nested structures.

5.2.2 Error Analysis

In this section, we addressed several aspects, including [1] the choice of pre-trained models (whether monolingual or multilingual pre-trained models perform better in terminology extraction); and [2] the effect of term length variants on the extraction ability of our classifier.

5.2.2.1 Monolingual vs. Multilingual Pre-trained Models

We evaluated the performance using monolingual language models, including XLNet[2] (English), CamemBERT[3] (French), and DutchBERT[4] (Dutch), compared with a multilingual model, XLMR[5], which was pre-trained on more than 100 different languages and fine-tuned for the downstream ATE task. The selection of monolingual models is based on their superior performance in the empirical evaluation of various monolingual and multilingual transformer-based models on the extraction of cross-domain monolingual sequence labeling terminology as shown in Section 4.

[2]xlnet-base-cased
[3]camembert-base
[4]GroNLP/bert-base-dutch-cased
[5]xlm-roberta-base

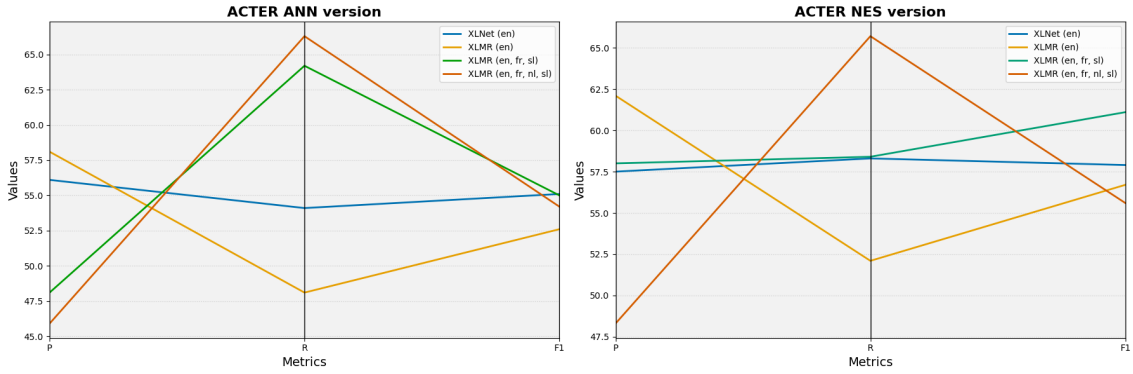


Figure 5.9: Performance of monolingual pre-trained classifier fine-tuned on English test language vs. multilingual one fine-tuned on the test language and multiple languages in ACTER.

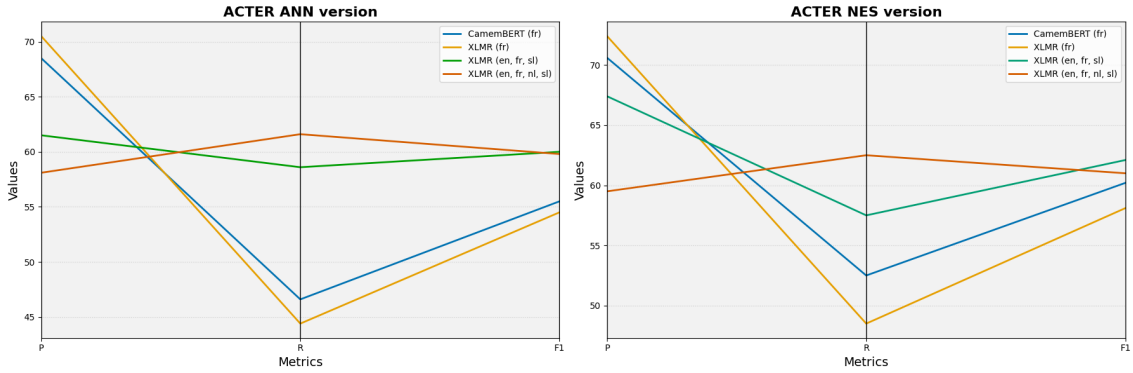


Figure 5.10: Performance of monolingual pre-trained classifier fine-tuned on French test language vs. multilingual one fine-tuned on the test language and multiple languages in ACTER.

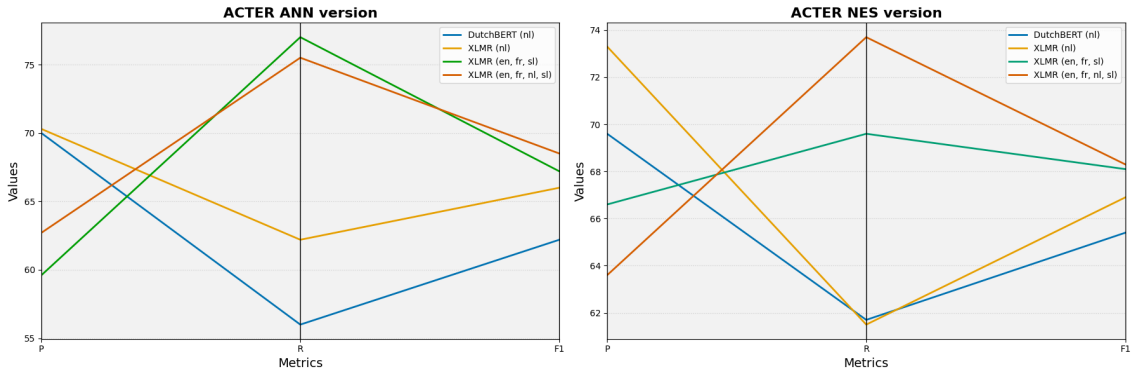


Figure 5.11: Performance of monolingual pre-trained classifier fine-tuned on Dutch test language vs. multilingual one fine-tuned on the test language and multiple languages in ACTER.

The results obtained with the monolingual models showed slightly higher performance in the specific language for which they were trained. Note that in our empirical studies, monolingual models (e.g., CamemBERT) performed better in a specific language (e.g., French) than other models pre-trained in that specific language (e.g., French) and than fine-tuned models in cross-lingual or multilingual learning, which can be attributed to several factors. Firstly, monolingual pre-trained models were trained exclusively in one

language, allowing them to optimize and specialize entirely in the linguistic structures, nuances, and vocabulary of that language. In contrast, multilingual pre-trained models had to reconcile the learning of multiple languages, resulting in less effective optimization for a single language. Secondly, the former models had a vocabulary and tokenization scheme specifically tailored to French, which better captured the unique morphological and syntactic features of the language. The latter models, on the other hand, used a common vocabulary that captured the specificities of each language less efficiently, especially for languages with rich morphology or specific characters. Despite the good performance of the monolingual model in the language in question, it was less competitive for other languages.

However, when applied in a cross-lingual context (e.g., fine-tuning XLNet on an English corpus and prediction on a French test set), the performance was significantly lower than that of the multilingual pre-trained model (e.g., XLMR). While the difference between the language-specific and multilingual models was small, the multilingual models trained with XLMR on the datasets of multiple and all languages outperformed the monolingual models by a small amount. Therefore, in order to consider and support multiple languages simultaneously, we decided to use XLMR as a benchmark model for all four languages in the ACTER and RSDO5 corpora to validate our hypotheses.

5.2.2.2 The Impact of Term Length

To determine whether the term length affects the models' performance, we calculated precision and recall for terms of length $k = \{1, 2, 3, 4, \geq 5\}$ when our classifiers predicted on the test set.

The ACTER dataset

We reported the precision and recall of the token classifier using BIO and NOBI annotation regimes in Figures 5.12, 5.13, and 5.14. When using the BIO scheme, the best model proved to be good at predicting terms containing up to four words for English and Dutch and up to three words for French texts in ACTER corpora. A strong correspondence between F_1 and the number of predicted candidate terms was found where the number of predicted candidate terms likely corresponded to the situation in the training data.

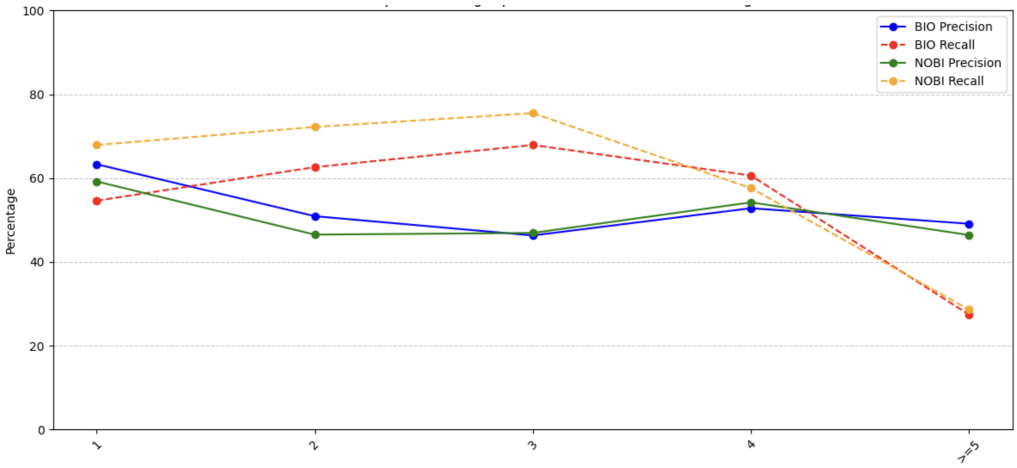


Figure 5.12: Performance in P and R per term length per domain in English ACTER test set.

The best models trained with the NOBI annotation regime demonstrated the same behavior as the models trained with the BIO annotation regime. They performed well in predicting terms with up to four words for English and Dutch and up to three words for

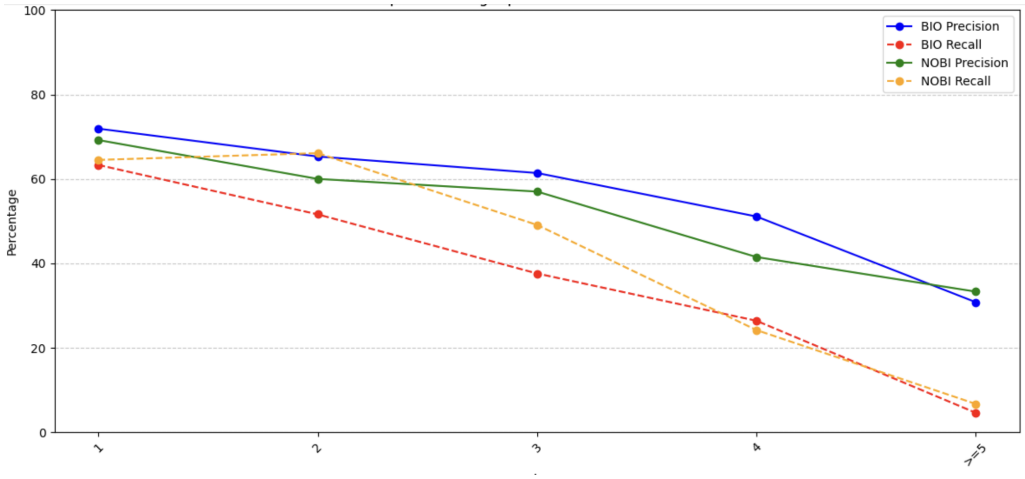


Figure 5.13: Performance in P and R per term length per domain in French ACTER test set.

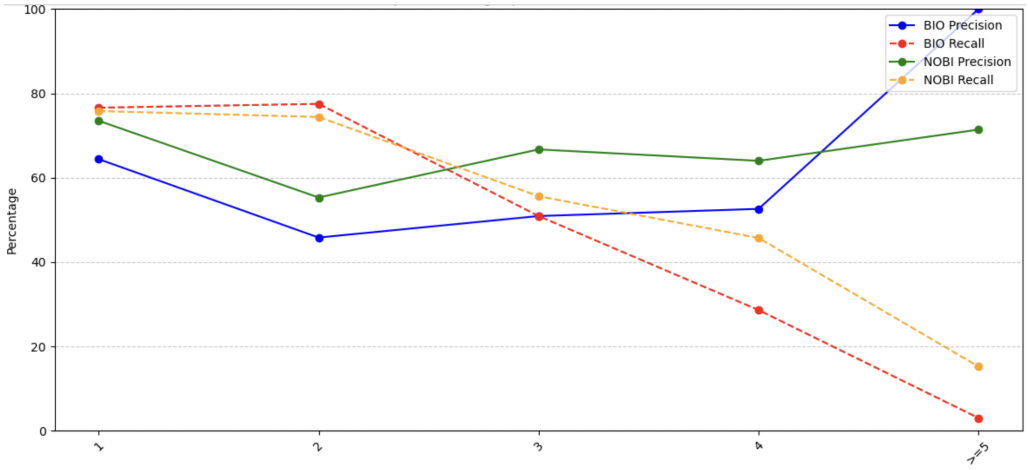


Figure 5.14: Performance in P and R per term length per domain in Dutch ACTER test set.

French texts in ACTER corpora. Although we expected that the NOBI annotation scheme would improve the model's ability to predict short, single nested terms, the classifiers trained with NOBI annotations also performed better on multi-word terms than those trained with the BIO regime, as long as nested terms constituted a reasonable proportion as in the ACTER corpora. We observed an improvement in recall for terms of all lengths, even for terms containing five words or more. There seemed to be some signal in the occurrence of nested terms within multi-word terms that caused the model to identify longer terms better. We assume that this effect is a combination of [1] the better identification of single-word terms due to a larger training set (both nested and independent single-word terms) and [2] nested terms acting as a kind of anchor that the model exploits to more easily identify multi-word terms around these nested terms. Further experimentation and analysis are required to fully understand this phenomenon.

In addition, a noticeable trend in most scenarios was that the NOBI regime decreased precision compared to the BIO regime. This appeared to be related to the number of terms predicted. Indeed, we observed that precision often decreased when the number of predicted terms is higher, i.e., the BIO regime predicts 1,009 single-word terms with a precision of 63.3% for the English dataset, while the NOBI regime predicts 1,341 terms

with a precision of 59.2%. A similar but opposite trend was observed for the Dutch NOBI regime, which predicted 1,738 terms with a precision of 73.5%, while the BIO regime yielded 2005 terms with a precision of 64.4%.

Table 5.10: A comparison of the performance between the BIO and NOBI regimes on the entire dataset, single-word (SWU), and multi-word (MWU) terms.

	BIO			**NOBI**		
	P	**R**	**F_1**	**P**	**R**	**F_1**
All terms	58.1	48.1	52.6	57.5	48.6	52.7
SWU terms	65.0	45.9	53.8	61.6	51.5	56.1
MWU terms	53.8	50.0	51.8	54.2	46.3	49.9

We also performed a detailed comparison of the monolingual BIO and NOBI results for the English dataset in Table 5.10. The NOBI regime produced a marginal improvement in F_1 and recall but had slightly lower precision. Overall, the algorithm predicted 1,956 candidates when using the BIO scheme and 1,996 when using the NOBI scheme. Of these, the BIO regime produced 751 one-word (SWU) terms and 1,205 multi-word (MWU) terms, while the NOBI regime produced 889 SWU terms and 1,107 MWU terms. As can be seen in Table 5.10, NOBI resulted in better recall of one-word terms (51.5% vs. 45.9%), which led to an overall improvement in F_1 (52.7% vs. 52.6%). This did not lead to an improvement in precision on SWU terms, but - perhaps surprisingly - to higher precision on MWU terms, which could be due to the NOBI regime favoring one-word terms (due to their higher proportion in the training set): This resulted in a lower number of higher quality MWU terms being predicted.

The RSDO5 dataset

The results for the RSDO5 dataset visualized from Figure 5.15 to 5.18 were obtained by employing the best-performing model in F_1 for each specific test domain for both annotation regimes, which were [1] training on *veterinary* and *chemistry*, validating on *biomechanics*, and testing on *linguistics* domain; [2] training on *linguistics* and *biomechanics*, validating on *chemistry*, and testing on *veterinary* domain; [3] training on *linguistics* and *veterinary*, validating on *biomechanics*, and testing on *chemistry* domain; [4] training on *linguistics* and *chemistry*, validating on *veterinary*, and testing on *biomechanics*.

Despite not improving in extracting single-word nested terms, the RSDO5 results were similar to those obtained with the ACTER corpora regarding the length of the candidate terms predicted by the classifier. They showed that the models for all four domains of the Slovenian corpus were able to predict short terms with up to three words well. The best model applied to the test domain *linguistics*, also performed relatively well in predicting longer terms, achieving 75.0% precision and 31.0% recall for terms with at least five words. Despite the relatively high precision in predicting long terms in the test domains *veterinary* and *biomechanics*, the recall was quite low, most likely due to the small number of longer terms in the dataset used to train the models. When predicting the domain *chemistry*, there were no correct predictions of terms with more than five words.

The NOBI regime often led to lower precision compared to BIO. Similarly to our findings on the ACTER dataset, this seemed to be related to the number of predicted terms. In general, the higher the number of predictions, the lower the precision (if the number of predicted terms is high enough - this trend was less noticeable for longer terms, of which there were only a few in the corpus). There were some exceptions, such as the *chemistry* domain, where the NOBI regime resulted in 909 predicted one-word terms with a precision of 61.4%, compared to 943 terms with a precision of 61.5% for the BIO regime,

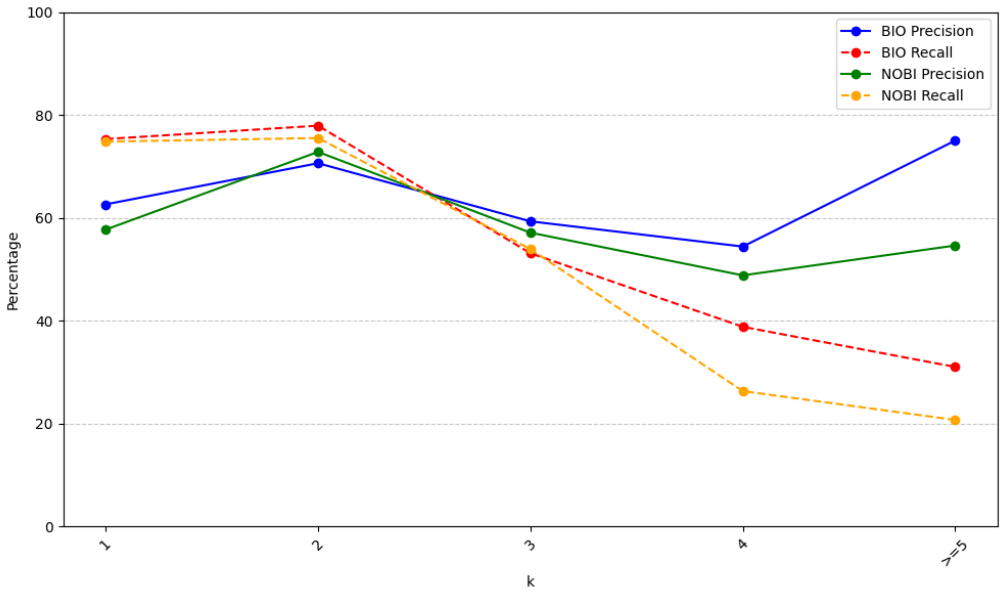


Figure 5.15: Performance in P and R per term length per domain in RSDO5 Linguistics test set.

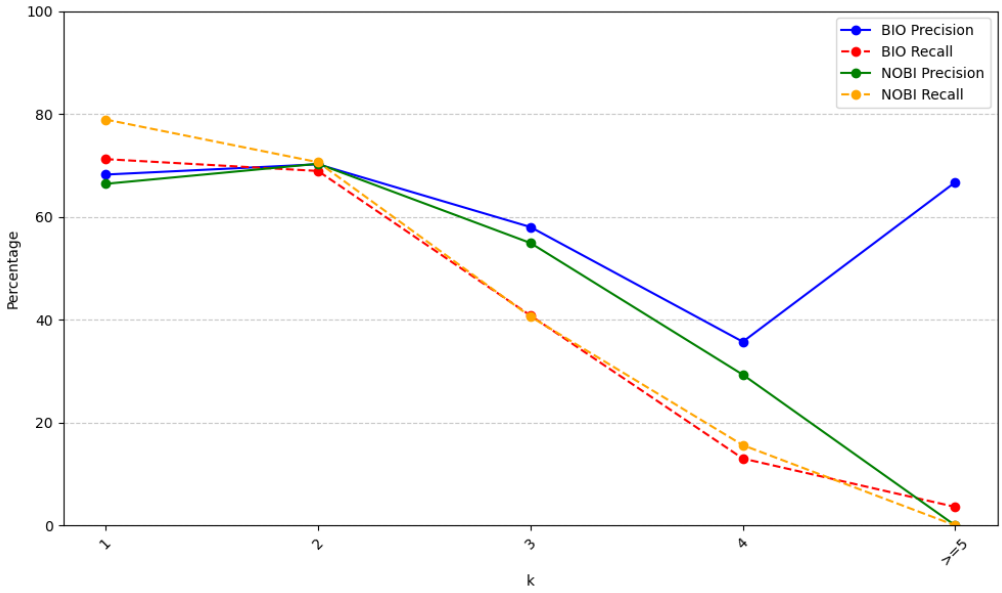


Figure 5.16: Performance in P and R per term length per domain in RSDO5 Veterinary test set.

and the *veterinary* domain, where the NOBI regime produced 2,111 two-word terms ($k = 2$) with a precision of 70.3%, while the BIO regime predicted 2,062 terms with a precision of 70.2%.

As mentioned above and in previous work (H. Tran et al., 2022) for the BIO regime, the very common error that both the BIO and NOBI models made was that they incorrectly predicted a shorter term that was nested in the correct term of the gold standard because the corpus contained nested terms. In contrast, the model sometimes produced incorrect predictions that contained the correct nested terms. However, it turned out that the NOBI annotation partially reduced the impact of these two mentioned error patterns and

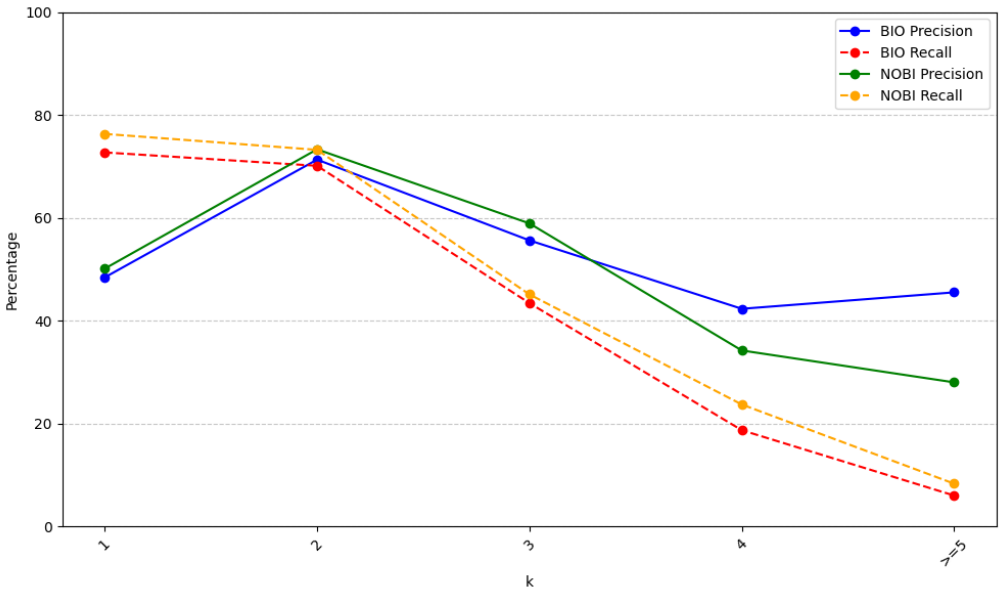


Figure 5.17: Performance in P and R per term length per domain in RSDO5 Biomechanics test set.

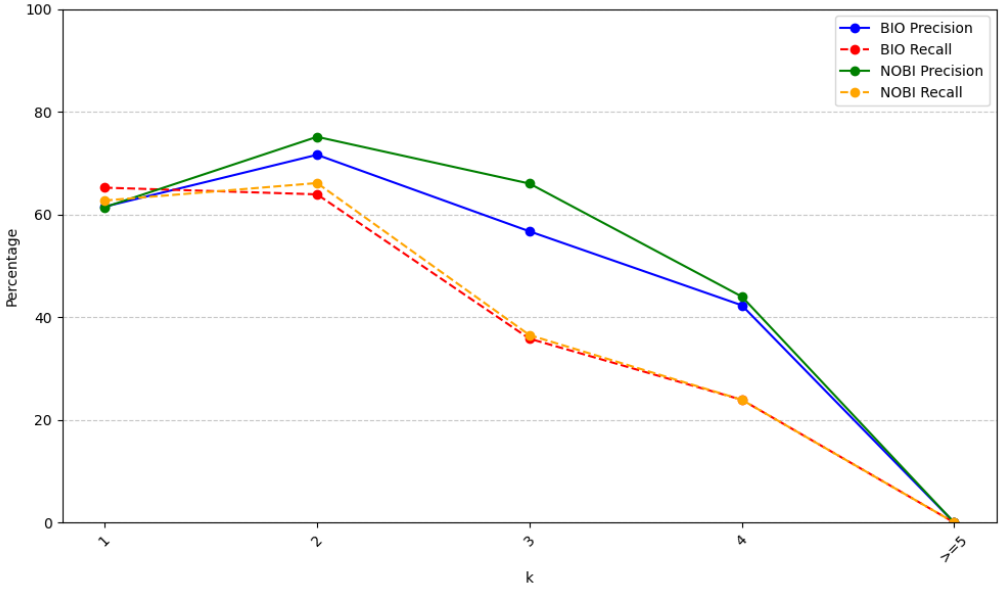


Figure 5.18: Performance in P and R per term length per domain in RSDO5 Chemistry test set.

improved the overall recall compared to the BIO benchmark.

5.3 Discussion

To summarize, we proposed a new NOBI annotation regime, that increased the predictive power of classifiers compared to the classical BIO mechanism by experimenting with multi-domain corpora, namely the ACTER and RSDO5 datasets, in monolingual, cross-lingual, and multilingual learning. The improvements thanks to the NOBI annotation regime were

visible for the dataset (e.g., as shown in ACTER corpora) in which the number of nested terms was significant enough, and also visible in the identification of multi-word terms, most likely due to the improvement of single-word terminology extraction and the use of single-word terms as anchors to correctly identify multi-word terms. The results confirmed the hypothesis H2.1: *"An annotation regime that captures additional information with regard to nested terms, improves the performance of token-based terminology extraction."*, which demonstrated the potential of the new annotation regime to improve the extraction of nested terms and the promising impact of cross-lingual and multilingual cross-domain learning in transferring from rich to less rich languages. The token classifiers that used NOBI for data annotation were published in the Terminology Extraction collection[6] on HuggingFace for practical use.

In the future, we will test the potential of our proposed NOBI mechanism on similar sequence labeling extraction tasks in other domains (e.g., keyword extraction, NER). In addition, we plan to investigate the integration of active learning into our current approach to improve the results of the automated method by dynamically adapting it according to human feedback.

[6]https://huggingface.co/collections/tthhanh/terminology-extraction-ate-66a26e41d723c565bbb8922f

Chapter 6

Generative Approaches for ATE Tasks

This chapter focuses on the third main hypothesis of this dissertation, namely **H3: Terminology Extraction Benefits from Generative Models** and more specifically the following four specific hypotheses concerning terminology extraction from the perspective of generative models as follows:

- **[H3.1] Terminology Extraction as Seq2Seq Classification Tasks:** *"Token classification models outperform Seq2Seq models on terminology extraction task."*

- **[H3.2] Large Language Models (LLMs) as Instructors for Terminology Extraction:** *"Large-scale language models with few-shot demonstration prompting using generative output formats leads to slightly lower performance, but avoids the need for extensive data annotation."*

- **[H3.3] The Domain Is Important for Automatic Terminology Extraction in the Era of LLMs :** [1] *"When employing LLMs for terminology extraction, few-shot demonstration prompting with self-verification allows us to predict terms without needing explicit information about the domain of the examples. This works for examples within the same domain as well as across different domains."*; [2] *"Using LLMs for few-shot demonstration prompting in cross-lingual transfer, with self-verification, allows effective transfer of knowledge from well-represented languages to less-represented ones."*

  In detail, we present our research on terminology extraction as a Seq2Seq task (H3.1) with the *templATE* classifier as described in Section 6.1.1, followed by the introduction of prompt engineering with LLMs (e.g., *ChatGPT*, *Llama-2*), named *promptATE*, using three forms of output designs (H3.2) for prompting (see Section 6.1.2). The results of Section 6.1.2 were considered in complementary experiments to the next empirical studies to find the optimal configurations of *LlamATE* (H3.3), a framework for testing the impact of domain specificity on the ATE task introduced in Section 6.1.3. For each of the sections from 6.1.1 to 6.1.3, the related publications are aligned. The results in comparison with the token classifier (so-called *iobATE*) with further error analysis are discussed in Section 6.2 before jumping into the conclusion in Section 6.3.

6.1 Models

We present our research on the applicability of closed-source (e.g., *ChatGPT*) and open-source (e.g., *Llama-2*) LLMs as *promptATE* classifier (see Section 6.1.2) to the ATE task by comparing them with two benchmarks where we consider ATE as a sequence labeling (*iobATE* classifier) and sequence-to-sequence (Seq2Seq) ranking task (*templATE* classifier) introduced in Section 6.1.1, respectively. The results in Section 6.1.2 were then considered in complementary experiments to the next empirical studies, in which we opted for tuning the best configuration for *LlamATE*, a framework for testing the impact of domain specificity on ATE when using in-context learning (ICL) prompts in open-sourced reinforcement learning with human feedback (RLHF) models in Section 6.1.3.

6.1.1 Sequence-to-Sequence (Seq2Seq) Models

We consider terminology extraction a sequence-to-sequence (Seq2Seq) task, which is similar to machine translation. Inspired by the work of González-Gallardo et al. (2023) for the NER task, we propose *templATE* where the original sentence is the source sequence and the templates filled by the candidate term span are the target sequence during training.

This part resulted in the following publication:

- (Accepted) (H. T. H. Tran, González-Gallardo, Delauney, et al., 2024) **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Julien Delaunay, Jose Moreno, Antoine Doucet, and Senja Pollak. "*Is Prompting What Term Extraction Needs?*". 27^{th} International Conference on Text, Speech, and Dialogue (TSD 2024), 2024.

6.1.1.1 Task Formulation

Let:

- $X = \{x_1, ..., x_n\}$ be the source sentence, where x_i represents the i^{th} word in the sentence.
- $T = \{t_1, ..., t_m\}$ be a set of pre-defined templates used in the training process. Each template t_i represents a template with a mask symbol "$< MASK >$" which will be replaced by the candidate term span during training.
- $Y = \{y_1, ..., y_l\}$ be the target sequence, where y_j is the output generated by the model for the j^{th} template-term span pair.

The objective of the model is to learn a scoring function f that assigns a higher score to positive template-term span pairs than negative ones.
Training Stage:

- Positive Examples: For each gold standard term span $x_{i:j}$ identified in sentence X, a positive training example $(t_k, x_{i:j})$ is created, where t_k is the template containing the mask that aligns with the term span. The corresponding target label for this pair is set to 1, denoting a positive term. Mathematically, this can be expressed as:

$$y_{pos,(k,i:j)} = f(t_k, x_{i:j}) = 1 \tag{6.1}$$

- Negative Examples: To address the class imbalance issue, a subset of negative examples is generated. Here, each word x_i in the sentence is considered as a candidate term span. A negative example (t_k, x_i) is formed for each template t_k and word x_i.

The target label for these pairs is set to 0, denoting a non-term. Mathematically, this can be represented as:

$$y_{neg,(k,i)} = f(t_k, x_i) = 0 \tag{6.2}$$

Inference Stage:

During inference, the model assigns a score to each possible n-gram (i.e., sequence of n words) in the sentence X. Here, n can vary from 1 to 4.

- Term Score Calculation: For each n-gram $x_{i:i+n-1}$, the model calculates a score using the template that best aligns with the n-gram. This involves finding the template t_k that maximizes the score among all possible templates. The term score can be formulated as:

$$\text{score}(x_{i:i+n-1}) = \max_{t_k \in T} f(t_k, x_{i:i+n-1}) \tag{6.3}$$

- Term Identification: An n-gram $x_{i:i+n-1}$ is considered a term if its corresponding score is greater than a predefined threshold θ:

$$Term(X) = \{x_{i:i+n-1} \in X \mid \text{score}(x_{i:i+n-1}) > \theta\} \tag{6.4}$$

where θ is the threshold value.

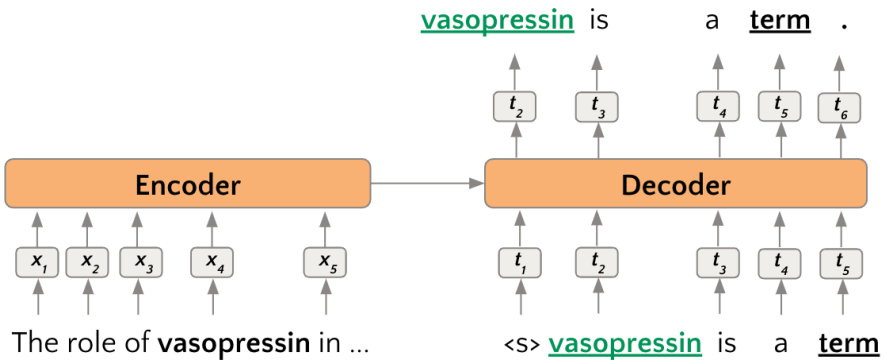
In short, this architecture aims to learn a scoring function f that discriminates between positive and negative training examples during training. This scoring function then assigns scores to n-grams in the sentence during inference, allowing for ATE based on a threshold.

6.1.1.2 Architecture

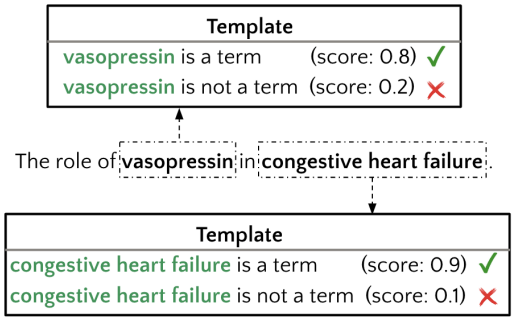
The task is formulated as a template-based (Seq2Seq) ranking problem for ATE task (we called it *templATE*), where the original sentence $X = \{x_1, ..., x_n\}$ is the source sequence, and the templates $T = \{t_1, ..., t_m\}$ filled by the candidate term span $x_{i:j}$ are the target sequence during training. The method contains the following steps:

1. Identify the gold standard terms in a sentence (e.g., *The role of* ***vasopressin*** *in* ***congestive heart failure****...*).
2. Create a positive template for gold standard terms: $< MASK >$ is a term. (e.g., ***vasopressin*** *is a term*; ***congestive heart failure*** *is a term*; ...).
3. Create a negative template for the rest: $< MASK >$ is not a term. (e.g., ***The*** *is not a term*; ***role*** *is not a term*; ***of*** *is not a term*; ***in*** *is not a term*;...).
4. **Training**: Feed into the mBART[1] (Tang et al., 2020) the sentence examples with positive templates (*"<X> is a term"*) and negative templates (*"<X> is not a term"*). Note that we only use 30% of negative templates generated to reduce imbalance. For example:
   - Sentence: *"The role of* ***vasopressin*** *in* ***congestive heart failure****"*.
   - Output: "*The* is not a term"; "*role* is not a term"; "*of* is not a term"; "*sentence* is not a term"; "*in* is not a term"; "*vasopressin* is a term"; "*congestive heart failure* is a term"; ...
5. **Inference**: Calculate the term score for each n-gram ($n = \{1, 2, 3, 4\}$). If the positive score is higher, consider it as a term.

[1] https://huggingface.co/facebook/mbart-large-50-many-to-one-mmt



(a) *templATE* training architecture.



(b) *templATE* inference.

Figure 6.1: *templATE* architecture.

6.1.1.3 Experimental Setup

We used mBART with 5 epochs, a batch size of 32, and a maxd sequence length of 70. The training and inference steps of the *templATE* approach are visualized in Figures 6.1a and 6.1b, respectively.

6.1.2 Prompt Engineering with LLMs

We propose *promptATE*, which uses the close-sourced ChatGPT's *gpt-3.5-turbo*[2] and the open-sourced *Llama 2-Chat* (i.e., *Llama 2-Chat-7B*[3], *Llama 2-Chat-13B*[4], and *Llama 2-Chat-70B*[5]) RLHF models to address the ATE task. RLHF plays an important role in aligning the model (e.g., *Llama 2-Chat*) with human preferences and values. The model was initially trained on a massive dataset of text and code (supervised fine-tuning). Then, a reward model was trained to predict how humans would rate different responses generated by the fine-tuned model (HF). This was done by collecting human ratings on a diverse set of prompts and responses. The language model was then fine-tuned using reinforcement learning (RL), where it learned to generate responses that maximized the reward from the reward model. This process helped the model to align its outputs with human

[2]https://platform.openai.com/
[3]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[4]https://huggingface.co/meta-llama/Llama-2-13b-chat-hf
[5]https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

preferences and avoid generating harmful or offensive content. Thus, RLHF helps *Llama 2-Chat* to become more helpful, informative, and harmless by learning from human feedback. The approach follows the general paradigm of in-context (i.e., few-shot) learning with a three-step procedure as in Figure 6.2, where [1] *Task Description* instructs *promptATE* to detect candidate terms using terminological knowledge; [2] *Few-shot Demonstrations* gives the model a few examples; and [3] *Input Sentence* indicates the input sentence while *promptATE*'s output is highlighted in green.

This part was conducted as part of the research in the following publication:

- (Accepted) (H. T. H. Tran, González-Gallardo, Delauney, et al., 2024) **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Julien Delaunay, Jose Moreno, Antoine Doucet, and Senja Pollak. "*Is Prompting What Term Extraction Needs?*". 27^{th} International Conference on Text, Speech, and Dialogue (TSD 2024), 2024.

6.1.2.1 Task Description

Given input sentence X, construct a $Prompt(X)$ to give a descriptive overview of the task with the following steps:

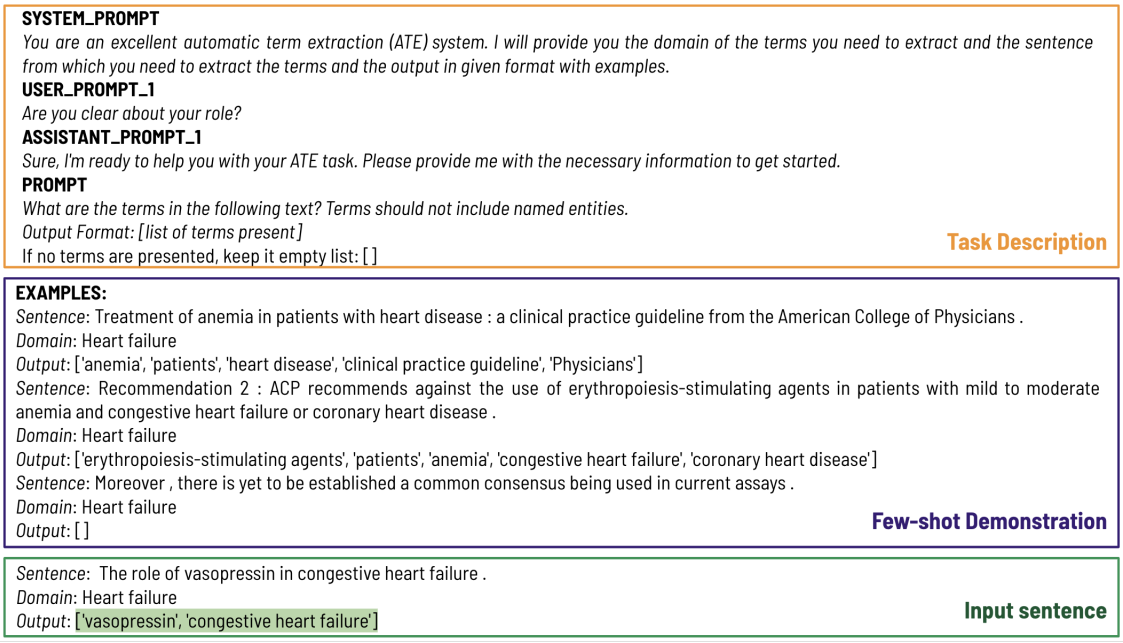


Figure 6.2: A complete prompt with the output format #2 for the ANN version (1) *Task Description* instructs *promptATE* to detect terms using terminological knowledge. (2) *Few-shot Demonstrations* give the model few-shot examples. (3) *Input Sentence* indicates the input sentence with the related domain while *promptATE*'s output is highlighted in green.

1. ***SYSTEM_PROMPT***: We define the role *promptATE* needs to assume with two sentences. First, *"You are an excellent automatic terminology extraction (ATE) system."* tells *promptATE* to produce the output using terminological knowledge. Second, *"I will provide you the domain of the terms you need to extract and the sentence from which you need to extract the terms"* indicates the input information, including the domain and sentence having domain-specific terms while *"and the output*

in the given format with examples." shows the position of few-shot demonstrations and marks the end of the description. This prompt is consistent throughout all the languages and data versions.

2. ***USER_PROMPT_1***: *"Are you clear about your role?".* triggers a response by the assistant explicitly asking for confirmation of the task comprehension.

3. ***ASSISTANT_PROMPT_1***: *"Sure. I am ready to help you with your ATE task. Please provide me with the necessary information to get started."* is the acknowledgment by *promptATE* but designed by the user only.

4. ***PROMPT***: This guideline prompt defines how *promptATE* should perform the ATE task. In the guidelines, we provided the requirements and the output format to guide *promptATE*'s responses for further processing.

6.1.2.2 Few-shot Prompting

In the few-shot demonstration, to regulate the output format for each test input, we provide examples during the task description phase. *promptATE* generates outputs mimicking the demonstration format. For example, in the *Few-shot Demonstrations* rectangle of Figure 6.2, the demonstration sequentially packs a list of examples, each consisting of both the input and output sequences. The demonstration is set up as follows: The first two examples contain terms, while in the last example, the input sentence does not include terms (see Figure 6.3). All the examples of sequences are from the test domain (*heart failure*) without any further information from the other three domains from ACTER corpora.

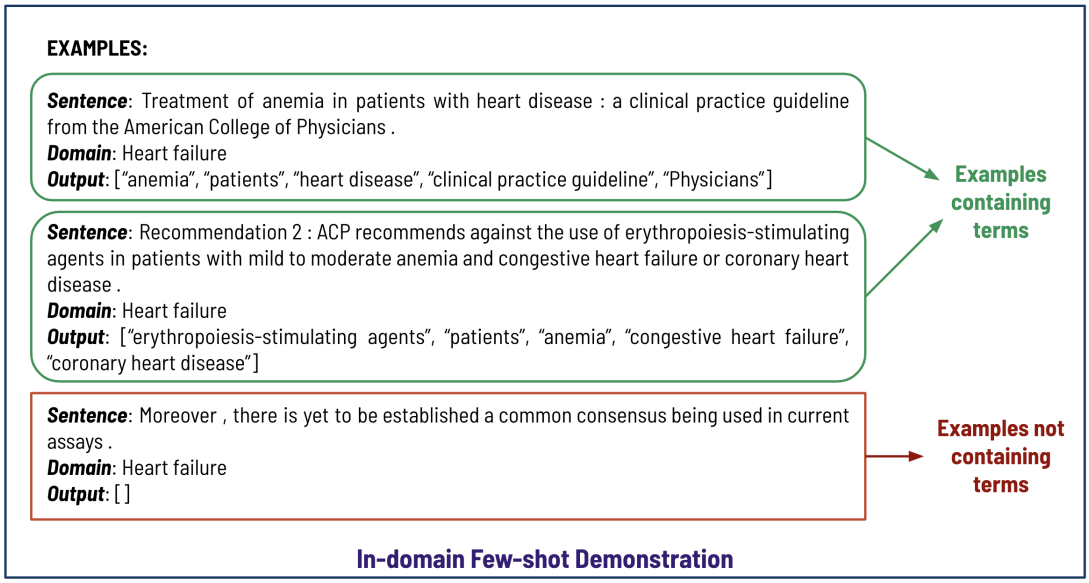


Figure 6.3: An example of how we set up in-domain Few-shot Demonstration.

The following output formats (OF) are tested: (1) *Sequence-labeling output (OF#1)*, where the output contains the information for each word label in the BIO annotation regime; (2) *List of candidate terms output (OF#2)*, which is the same format as our original gold standard; (3) *Generative output (OF#3)*, where we use unique tokens "@@" and "##" to surround the candidate terms.

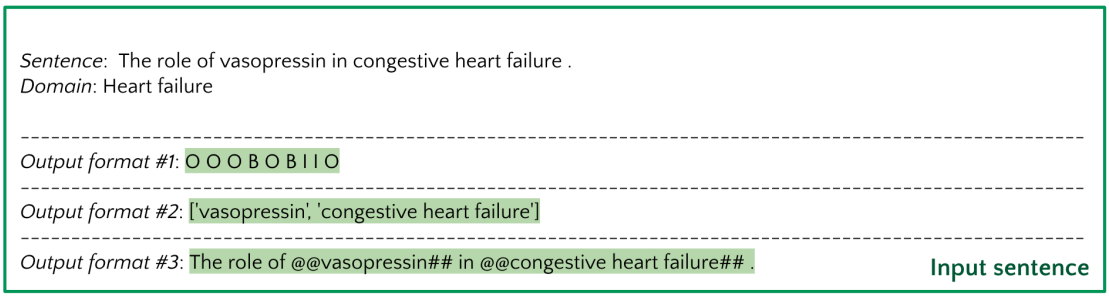


Figure 6.4: An example of three output designs.

6.1.2.3 ChatGPT vs. Llama 2-Chat

We focus on the capabilities of few-shot demonstrations, employing both the close-sourced ChatGPT (*gpt-3.5-turbo*) (Brown et al., 2020) and the open-sourced *Llama 2-Chat* (Touvron et al., 2023) models. While both exhibit remarkable language understanding and generation abilities, they employ divergent training mechanisms and prompting syntax. Thus, slight modifications are required in the prompt structure while preserving the overarching concepts. By doing so, we aim to evaluate how each model adapts to varying input cues and assess their respective adaptability in handling the same set of instructions. This set of experiments compares the performance of these RLHF models and underscores their flexibility and versatility in comprehending and generating content, even when their underlying architectures differ significantly. The results are presented in Section 6.2.

6.1.3 Lexical and Domain Specificity in the Era of LLMs

The domain has a great influence on the choice of terms in a collection of texts. A linguistic unit in a text can be considered a term (or not) depending on the domain of interest. However, how specialized or domain-specific this lexical unit needs to be before it is considered a term is far from a consensus. The concept of termhood was introduced by Kageura and Umino (1996) to indicate the degree of relationship between a lexical unit and certain concepts within a domain. Rigouts Terryn (2021) addressed this problem by defining "termhood" as a function of lexicon and domain specificity in her annotation guidelines. On the one hand, lexicon-specificity denotes if a lexical unit is only known by specialists or if it is part of a common language. On the other hand, domain-specificity denotes whether a lexical unit is relevant or unrelated to the researched domain. The combination of these two indicators results in four different types of classes: specific terms, common terms, out-of-domain terms, and no terms. This classification, which is far from perfect and has a significant degree of subjectivity, proves to be very useful and leads to more intuitive annotations.

With the evolution of LLMs as described in Section 6.1.2, in this section, we first discuss the importance of the domain for terminology extraction, followed by our preliminary studies on the optimal configuration for the prompting technique where LLMs are the instructors. These configuration components consider [1] the optimal LLMs size/version, [2] the optimal output format, [3] the optimal number of demonstrations, and [4] interactive or non-interactive demonstrations. Based on the results of preliminary studies, we propose *LlamATE*, an architecture for a few-shot prompting term extractor using Llama-2-Chat as the foundation instructor, with the corresponding experimental setup to evaluate the importance of domain and language in the era of LLMs.

This part resulted in the following publication:

- (Accepted) (H. T. H. Tran, González-Gallardo, Doucet, & Pollak, 2024) **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Antoine Doucet, Senja Pollak. "*LlamATE: Automated Term Extraction Using Large-scale Generative Language Models*". Computational Terminology Special Issue – Terminology, 2024.
- (Accepted) (H. T. H. Tran, González-Gallardo, Moreno, et al., 2024) **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Jose Moreno, Antoine Doucet, and Senja Pollak. "*Is Domain Important for Automatic Term Extraction in the Era of Large Language Models?*". Terminologie & Ontologie : Théories et applications (ToTh 2024), 2024.

6.1.3.1 Domain Relevance in Terminology Extraction

If term selection depends on the lexical capacity and expertise of the annotator, how would these factors influence an ATE system based on LLMs with in-context learning? LLMs are trained with billions or trillions of tokens (i.e., chunks of words) collected mainly from publicly available sources. This amount of training text gives the LLMs the ability to generate plausible-sounding text that follows the syntax and semantics of a given prompt. In addition, the LLMs are domain-independent due to the amount of different text. This enhances their versatility, allowing them to be used for numerous applications. However, domain-specific knowledge can be crucial in cases such as terminology extraction.

We find two analogies between the *termhood* (Kageura & Umino, 1996; Vezzani et al., 2020) as used by Rigouts Terryn (2021) in their annotation guidelines of the ACTER dataset and LLMs in-context learning for terminology extraction. First, lexicon-specificity can be represented with few-shot demonstrations that illustrate, independently of the domain, the lexicon of input sentences and the desired output list of terms. Second, domain-specificity can be induced into the prompt by explicitly declaring the domain of interest or letting the language model deduce it from the few-shot demonstrations. These two analogies result in four different prompt scenarios to verify the impact of lexical and domain in the ATE task within the era LLMs : [1] in-domain demonstrations with explicit domain, [2] cross-domain demonstrations with explicit domain, [3] in-domain demonstrations with implicit domain, and [4] cross-domain demonstrations with the implicit domain. In the next section, we describe how prompts are structured and detail the methodology we followed to test each scenario.

6.1.3.2 Preliminary Studies

The quality of the input prompt when querying an LLM system is essential to the quality of the response. The popularization of LLMs was followed by the development of prompt engineering, the aim of which is to design prompts that best fit the language model to maximize the quality of the generated text. As a rule of thumb, the more explicit the prompt, the better. Some examples of the task can be included to show the language model the sort of answers that are expected; this is known as a few-shot demonstration. Moreover, specifying certain criteria like the field or discipline that is relevant to the task narrows down the language model into a fixed text generation style.

Therefore, we carry out a series of complementary experiments, including the verification of [1] the size/version of *Llama-2-Chat*, [2] the optimal output format, [3] the optimal number of demonstrations, and [4] interactive or non-interactive demonstrations. We captured the optimal choice for each step, as highlighted in bold arrows in Figure 6.5.

Model Sizes/Versions: We evaluate the performance of different versions of *Llama-2-Chat*, including *Llama-2-Chat-7B*, *Llama-2-Chat-13B*, and *Llama-2-Chat-70B*. Table 6.1 demonstrates the characteristics of *Llama-2-Chat* regarding the number of parameters used

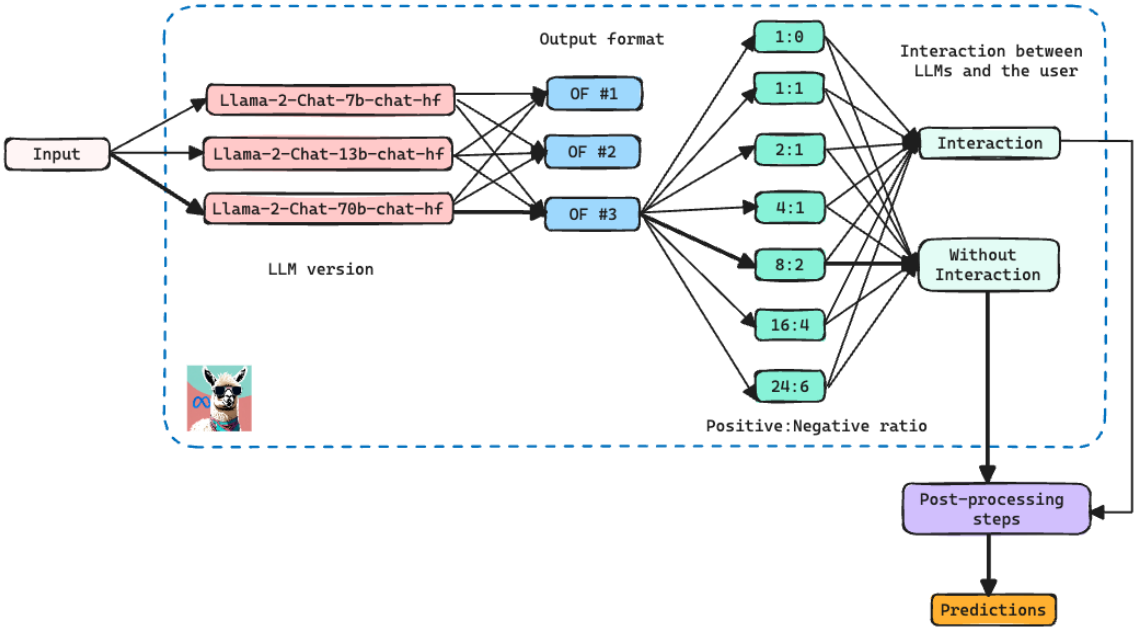


Figure 6.5: Our 4-step procedure to choose the optimal configuration in the in-domain few-shot prompting workflow. The bold arrow demonstrated the optimal path.

for pre-training, the training size (number of tokens), content length (CL), Grouped-Query Attention (GQA), and the percentage (%) of data the models contain in English, French, and Dutch, respectively, for pre-training.

Table 6.1: Characterization of models used in our experiments.

Models	# Params	CL	GQA	Content Length	English	French	Dutch
Llama-2-Chat-7B	7B	4k	X	2T	89.70	0.16	0.12
Llama-2-Chat-13B	13B	4k	X	2T	89.70	0.16	0.12
Llama-2-Chat-70B	70B	4k	✓	2T	89.70	0.16	0.12

Prompt's Output Designs: We investigate three common outputs of language models for terminology extraction and adapt them to our prompts' output format with in-domain few-shot demonstrations, as visualized in Figure 6.4. The output formats include (1) *OF #1*: Sequence-labeling output where the output contains the information for each word label in the BIO regime; (2) *OF #2*: List of candidate terms output which is the same format as our original gold standard; and (3) *OF #3*: Generative output where we use unique tokens "@@" and "##" to surround the candidate terms.

Optimal Number of Demonstrations: To study the behavior of *LlamATE* in a few-shot context, we simulate the few-shot context by providing the models with only a few annotated examples with and without interaction on the English *heart failure* test set. We verify the optimal number of demonstrations and the proportion of positive and negative examples that should be provided to *LlamATE* so that it maximizes the number of correct predictions. Furthermore, we evaluate whether *LlamATE* performs better when there exists an interaction between the LLMs and the user during the prompting process. In the few-shot demonstration, seven positive and negative ratios are tested, including 1:0, 1:1, 2:1, 4:1, 8:2, 16:4, and 24:6. We start with a positive example to ensure the LLMs

understand how to formulate the prediction outputs, which is not the case in the zero-shot demonstration, and we increase the number of examples at the same ratio based on the ratio of positive and negative sentence-level examples in the corpus.

Interaction and Without Interaction: Regarding the interaction between the LLMs and the user, for each ratio, we experiment with two scenarios where [1] we feed all the examples simultaneously as a demonstration (no interaction) and [2] we feed each of the examples sequentially as the interaction between the LLMs and the user as a demonstration (interaction).

6.1.3.3 Architecture

The advent of LLMs has significantly improved performance on several downstream tasks using two different strategies: fine-tuning and ICL. While the fine-tuning strategy involves initializing a pre-trained model and running additional training epochs on task-specific supervised data, ICL leverages the LLM's ability to generate text with only a few task-specific examples as demonstrations. On the one hand, training or fine-tuning LLMs specifically tailored to a particular task (e.g., terminology extraction) is challenging as it requires significant computational resources (X. Wang et al., 2023). On the other hand, we believe that the extensive world knowledge of LLMs offers promising insights to improve the processing of such extraction tasks.

As a result, in our research, we applied the ICL strategy to *Llama-2-Chat*, the RLHF version of the open-sourced auto-regressive LLMs released by Meta AI, which is tuned to human preferences for helpfulness and safety. Based on the results of the preliminary studies, we use the largest version of Llama-2-Chat, namely *Llama-2-Chat-70B*[6], trained between January 2023 and July 2023. Unlike other versions, it uses Grouped Query Attention (GQA) to improve the scalability of inference. The general workflow of our proposed *LlamATE* is shown in Figure 6.6.

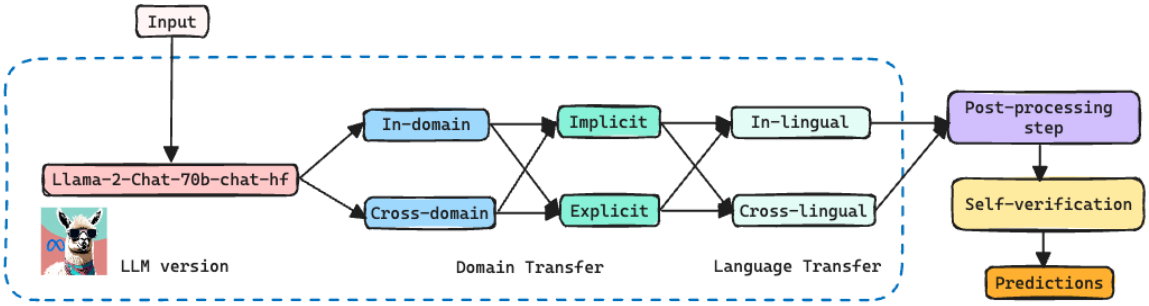


Figure 6.6: Our general *LlamATE* workflow for few-shot prompting term extractor.

Methodology Design: The quality of the response from the LLM system heavily depends on the quality of the input prompt, which, as discussed previously: As a rule of thumb, the more explicit the prompt, the better. We have therefore designed our prompt as follows (see example in Figure 6.7).

1. *Prompt language*: By default, we use prompt instruction in English for all scenarios, even for terminology extraction in another language (e.g., French), as English is the most ubiquitous language in all their training corpora (i.e., 89.7% of training data in *Llama-2-Chat* is in English).

[6]https://huggingface.co/meta-Llama/Llama-2-70b-chat

```
As an excellent automatic term extraction (ATE) system, extract the terms in the Heart Failure domain
given the following text delimited by triple backquotes. Named entities are not considered as terms.
                                                                                    Task Description

Examples of the output format:
Sentence: ```Treatment of anemia in patients with heart disease : a clinical practice guideline from
            the American College of Physicians .```
Domain: Heart failure
Output: "Treatment of @@anemia## in @@patients## with @@heart disease## : a @@clinical practice
         guideline## from the American College of @@Physicians## ."

Sentence: ```Recommendation 2 : ACP recommends against the use of erythropoiesis-stimulating agents
            in patients with mild to moderate anemia and congestive heart failure or coronary heart
            disease .```
Domain: Heart failure
Output: "Recommendation 2 : ACP recommends against the use of @@erythropoiesis-stimulating agents##
         in @@patients## with mild to moderate @@anemia## and @@congestive heart failure## or
         @@coronary heart disease## ."

                                          ...


Sentence: ```Moreover , there is yet to be established a common consensus being used in current
            assays .```
Domain: Heart failure
Output: "Moreover , there is yet to be established a common consensus being used in current assays ."

Sentence: ```Literature was reassessed in April 2013 , and additional studies were included .```
Domain: Heart failure
Output: "Literature was reassessed in April 2013 , and additional studies were included ."
                                                                                  Few-shot Demonstration

Sentence: ```<text>```
Domain: Heart failure                                                               Input Sentence
```

Figure 6.7: An example of the prompt we used to extract the candidate terms for the ANN version of the ACTER dataset in the explicit in-domain in-lingual settings.

2. *Task Description*: We define the role that the LLMs need to follow and specify the task the LLMs have to perform: *"As an excellent automatic terminology extraction (ATE) system, extract the terms in the Heart Failure domain given the following text delimited by triple backquotes."*. We provide an additional sentence to clarify the scope of the candidate terms. In the ANN version, we mention *"Named entities are not considered as terms."*, while in and in the NES version, we mention *"Named entities are considered as terms."*.

3. *Few-shot Demonstration*: In the context of the ATE task, a standard few-shot or k-shot sample refers to obtaining exactly a few (k) occurrences of examples. However, meeting this strict requirement can be challenging, especially when dealing with data that exhibits an imbalanced distribution between sentences containing terms and sentences that do not, both within a domain and across domains. To address this challenge, we draw inspiration from the work of Ding et al. (2021) and introduce a relaxation to the criterion. We conduct an empirical study to find the optimal number of positive and negative examples to demonstrate LLMs and use relaxation methods to permit a maximum of $1.2k$ occurrences in cross-domain k-shot scenarios. We provide examples that are appended to the task description phase to regulate the format of outputs for each test input. The demonstration sequentially packs a list of examples, each consisting of the input (sentence and/or domain) and the output sequences. The output sequences use the generative output format where we use unique tokens "@@" and "##" to encapsulate the candidate terms.

Note that for the demonstration, we opt to extract the candidate term, regardless

of the nature of the term. Thus, we have not distinguished between the ACTER term annotations, where specific terms, general terms, OOD terms, and NEs are distinguished.

Domain Transfer: We find two analogies between *termhood* and LLMs with ICL for the terminology extraction tasks. First, domain-specificity can be induced into the prompt by explicitly declaring the domain of interest or letting the language model deduce it from the few-shot demonstrations. Second, lexicon-specificity can be represented with few-shot demonstrations that illustrate the lexicon of input sentences and the desired output list of terms independently of the domain. These two analogies result in four different prompting scenarios, including [1] *Explicit in-domain*: In-domain demonstrations with explicit domain; [2] *Implicit in-domain*: In-domain demonstrations with implicit domain; [3] *Explicit cross-domain*: Cross-domain demonstrations with explicit domain; and [4] *Implicit cross-domain*: Cross-domain demonstrations with implicit domain. With these scenarios, we verify the influence of the domain on the predictive performance of *LlamATE* when designing prompts with few demonstrations to help LLMs capture the candidate terms in the correct formats.

Language Transfer: We evaluate the capability of *LlamATE* to apply the knowledge learned from the over-represented language (i.e., English, which accounts for 89.70% of *Llama-2-Chat* training data) in the ATE task to another unseen under-represented language in the Llama model (i.e., French and Dutch). Therefore, for both French and Dutch, we use the English prompts where the demonstrations are the examples extracted from the English corpus and combine this setting with the domain transfers mentioned above for both the ANN and NES versions. In this scenario, we examine how well *LlamATE* performs without the language-specific examples and how good the knowledge transfer between different languages is.

Postprocessing Steps: More than 95% of the predictions returned by *LlamATE* correspond to the original sentences with the candidate terms encapsulated by the symbols "@@" and "##". However, despite not asking for an explanation or any further information, *LlamATE* occasionally returns the encapsulated sentence and [1] further explanation; [2] a summary of the candidate list in the form of a bulleted list with or without encapsulated symbols, or [3] a small note for a conclusion. We thus add a postprocessing step to normalize the predicted outputs. We skip the unasked details that *LlamATE* provides to avoid potential noise (as the extra information might cover hallucinated candidate terms) and keep only the answered sentence. Then, we filter the candidate terms using regex and formulate them as a list of candidate terms for each sentence.

Self-verification: LLMs significantly suffer from *hallucination* or overprediction issues, even with demonstrations (Xu et al., 2024). To address this issue, we have developed a self-verification strategy. Given the candidate terms extracted by *LlamATE*, we ask the LLMs to further verify whether the extracted entity is correct by responding with *YES* or *NO* with and without *Explanation*.

- *YES/NO verification*: We ask the model to self-verify if the list of extracted candidate terms it provides is correct given the sentence and each term in the list. By doing so, the step helps us filter out the false positives. Sharing the same mechanism as the main prompt, the format of our self-verification prompt is exemplified in Figure 6.8.

- *YES/NO verification with Explanation*: Similar to *YES/NO verification*, we ask the model to check the list of extracted candidate terms but elaborate on its answer by asking to provide useful evidence for humans to verify it. This self-verification prompt is depicted in Figure 6.9.

In both cases, we pack multiple demonstrations with the same number of positive and

```
As an excellent automatic term extraction (ATE) system, verify whether a given phrase is a term in
the Heart Failure domain in the given sentence by answering YES or NO.
Named entities are not considered as terms.                                    Task Description

Examples of the output format:
Sentence: ```Treatment of anemia in patients with heart disease : a clinical practice guideline from
          the American College of Physicians .```
Phrase: anemia
Output: YES

Sentence: ```Recommendation 2 : ACP recommends against the use of erythropoiesis-stimulating agents
          in patients with mild to moderate anemia and congestive heart failure or coronary heart
          disease .```
Phrase: erythropoiesis-stimulating agents
Output: YES

                                          ...


Sentence: ```Treatment of anemia in patients with heart disease : a clinical practice guideline from
          the American College of Physicians .```
Phrase: Treatment
Output: NO

Sentence: ```Recommendation 2 : ACP recommends against the use of erythropoiesis-stimulating agents
          in patients with mild to moderate anemia and congestive heart failure or coronary heart
          disease .```
Phrase: Recommendation
Output: NO

                                                                    Few-shot Demonstration

Sentence: ```<text>```
Phrase: <candidate terms>                                                  Input Sentence
```

Figure 6.8: An example of the YES/NO verification for the ANN version.

negative examples in the verification prompt. Demonstrations are followed by the test example, and fed to the LLMs to obtain the output.

6.1.3.4 Experimental Setup

We use the largest version (70B) of *Llama-2-Chat*, developed by Meta AI, the most powerful model of the Llama sub-series as of mid-April 2024 via the API endpoint of HuggingFace. In our experiments, the Llama-2-Chat is used only as a predictor, we only query the model, and no additional fine-tuning steps have been realized. Based on the sentence length in the corpora, we set the output length of the model to 1,024 tokens. With regard to experimental reproducibility, we set the temperature parameter to 0.1 (close to zero, but not zero). This parameter must be a strictly positive floating point number because if the temperature is exactly zero, this would effectively mean dividing by zero or multiplying by infinity, which would lead to invalid probabilities. This makes the sampling of the model almost deterministic and selects the most likely token with very high probability. All experiments were performed on CPUs in a Macbook Pro Ventura 13.6.5, 2.9 Quad-Score Intel i7, 16 GB memory.

6.2 Results

In this section, we define the evaluation metrics in Section 6.2.1 to measure the performance of different approaches introduced earlier in Chapter 6 compared to the baseline as described in Chapter 4.2.2. We then present general observations and discussions on the results obtained in Section 6.2.2. A more detailed error analysis is provided in Section

```
As an excellent automatic term extraction (ATE) system, verify whether a given phrase is a term in
the Heart Failure domain in the given sentence by answering YES or NO with explanation.
Named entities are not considered as terms.                                    Task Description

Examples of the output format:
Sentence: ```Treatment of anemia in patients with heart disease : a clinical practice guideline from
          the American College of Physicians .```
Phrase: anemia
Output: YES
Explanation: Anemia is a condition in which the body does not have enough healthy red blood cells.
          Red blood cells provide oxygen to body tissues.

Sentence: ```Recommendation 2 : ACP recommends against the use of erythropoiesis-stimulating agents
          in patients with mild to moderate anemia and congestive heart failure or coronary heart
          disease .```
Phrase: erythropoiesis-stimulating agents
Output: YES
Explanation: Erythropoiesis-stimulating agents are drugs that stimulate the production of red blood
          cells.

                                          ...


Sentence: ```Recommendation 2 : ACP recommends against the use of erythropoiesis-stimulating agents
          in patients with mild to moderate anemia and congestive heart failure or coronary heart
          disease .```
Phrase: Recommendation
Output: NO
Explanation: Recommendation is a suggestion or proposal as to the best course of action, especially
one put forward by an authoritative body.
                                                                  Few-shot Demonstration

Sentence: ```<text>```
Phrase: <candidate terms>                                                  Input Sentence
```

Figure 6.9: An example of the YES/NO verification with explanation for the ANN version.

6.2.3 to gain deeper insights into model behavior.

6.2.1 Evaluation Metrics

While we evaluate *templATE* and *promptATE* with only evaluation metrics, we assess the performance of the in-context learning (ICL) prompt in *LlamATE* using both evaluation and environmental metrics as below:

- *Evaluation metrics*: The performance of each term extractor is measured by strictly comparing the aggregated list of candidate terms identified across the entire test set against the manually designated gold standard list of terms, using precision (P), recall (R), and F_1-score (F_1). These evaluation metrics were the same as for the experiments in Sections 4 and 5.

- *Environmental metrics*: We use Green Algorithms[7] v2.2 to estimate the carbon footprint of each experiment, based on factors such as runtime, computing hardware, and location where electricity used by our computer facility was produced.

6.2.2 Quantitative Results

We reported the performance of our methods for each of the three hypotheses described above on ACTER corpora using precision, recall, and F_1 as evaluation metrics for quantitative results and Green Algorithms as the environmental metric.

[7] http://calculator.green-algorithms.org/

6.2.2.1 Optimal Configurations

Model versions. In Table 6.2, we evaluated the performance of these output formats with different versions of *Llama-2-Chat* compared to close-sourced ChatGPT's *gpt-3.5-turbo* in the in-domain few-shot setting and our fine-tuned token classifier using different LLMs given the same 7B hyperparameters (same setting with XLMR benchmark) where we highlight in grey the best performing format for each version.

Table 6.2: Evaluation in performance of different output formats on the Heart Failure test set with in-domain demonstrations (OF = Output Format). Bold font highlights the best performance in each evaluation metric for each language and domain version.

Settings	ANN versions									NES versions								
	English			French			Dutch			English			French			Dutch		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Fine-tuned token classifier$_{Llama-2-Chat-7B}$																		
Token classifier	35.2	46.1	39.9	35.7	53.3	42.8	36.0	66.9	46.8	36.4	57.1	44.5	39.1	51.7	44.5	42.4	64.6	51.2
In-domain *LlamATE*$_{Llama-2-Chat-7B}$																		
OF #1	12.4	4.8	6.9	7.5	9.3	8.3	19.2	14.4	16.5	17.3	7.3	10.3	8.4	11.0	9.5	16.6	23.8	19.6
OF #2	40.4	62.6	**49.1**	36.3	59.2	45.0	40.4	73.1	52.0	42.9	63.4	51.2	36.0	61.6	45.4	40.3	75.6	52.6
OF #3	40.3	26.8	32.2	**58.5**	23.4	33.4	53.8	41.6	46.9	45.0	32.5	37.7	**52.1**	34.5	41.5	48.8	52.3	50.5
In-domain *LlamATE*$_{Llama-2-Chat-13B}$																		
OF #1	12.1	1.7	3.0	11.2	6.6	8.3	25.6	5.9	9.6	25.9	2.4	4.4	8.4	5.3	6.5	23.5	5.9	9.4
OF #2	35.0	63.4	45.1	38.4	59.2	46.6	43.3	75.0	54.9	38.4	66.1	48.6	33.8	60.2	43.3	41.4	73.6	53.0
OF #3	40.0	36.9	38.4	41.0	48.7	44.5	46.1	56.2	50.7	40.3	47.5	43.6	35.7	49.4	41.4	47.9	49.1	48.5
In-domain *LlamATE*$_{Llama-2-Chat-70B}$																		
OF #1	15.6	5.7	8.3	4.6	3.9	4.2	23.7	8.2	12.2	21.4	4.7	7.7	7.9	9.5	8.6	18.7	17.5	18.1
OF #2	36.8	65.9	47.2	38.0	**64.8**	47.9	42.3	74.8	54.0	39.9	67.2	50.1	33.2	61.8	43.2	41.1	74.8	53.1
OF #3	**46.4**	50.0	48.1	47.1	51.4	**49.2**	50.5	67.3	57.7	**48.3**	54.9	**51.4**	40.8	57.7	47.8	53.1	57.9	55.4
In-domain *ChatGPT*$_{gpt-3.5-turbo}$																		
OF #1	10.8	14.4	12.3	11.3	11.6	11.4	18.3	14.1	15.9	10.3	13.1	11.5	10.8	12.0	11.4	14.8	13.2	14.0
OF #2	26.6	**67.6**	38.2	28.5	67.0	40.0	36.8	**79.6**	50.3	29.2	**69.2**	41.1	27.9	**66.8**	39.4	39.8	**78.5**	52.8
OF #3	39.6	48.3	43.5	45.5	50.8	48.0	**61.1**	56.6	**58.8**	39.8	53.1	45.5	44.7	54.4	**49.1**	**63.6**	60.6	**62.1**

Regarding the performance of prompting against the fine-tuned token classifier sharing the same LLMs (e.g., *Llama-2-Chat-7B*), the *OF #2* outperformed the token classifier in most cases. Specifically, prompting with *OF #2* surpassed in F_1 for all languages and versions, and *OF #3* provided competitive performance compared to the token classifier using the same LLMs. This could be due to the fact that the number of records for each language is limited to fully utilize the power of LLMs in fine-tuning. Moreover, the context we provided to the LLMs via prompt engineering could incorporate the surrounding text and domain knowledge to provide context and guide the model. This helped the model understand the role of each word within the candidate term without training and required fewer annotated examples for the model. Since ICL worked well enough and was computationally cheaper than fine-tuning, we opted for prompting.

With respect to different output formats, the results showed a significant difference in performance depending on the output format. *LlamATE* struggled with low precisions and recalls in sequence labeling (*OF #1*) compared to the other two formats. This indicated the discrepancy between the semantic labeling task and the text generation task for which *Llama-2-Chat* was trained. Meanwhile, the formats *OF #2* and *OF #3* had up to 8 times higher F_1 than *OF #1* regardless of the language and model size. On the one hand, larger models with more parameters (e.g., *Llama-2-Chat-70B*) tended to be better able to capture complex linguistic relationships and generalize better to unseen data than models with fewer parameters (e.g., *Llama-2-Chat-7B*, *Llama-2-Chat-13B*). This could be advantageous for cross-lingual transfer, as the model had to learn transferable knowledge from the source language to the target language. On the other hand, larger models with

a larger vocabulary allowed the model to process a wider range of words and concepts, which was crucial for cross-lingual transfer, as the source and target languages may have different vocabulary sets.

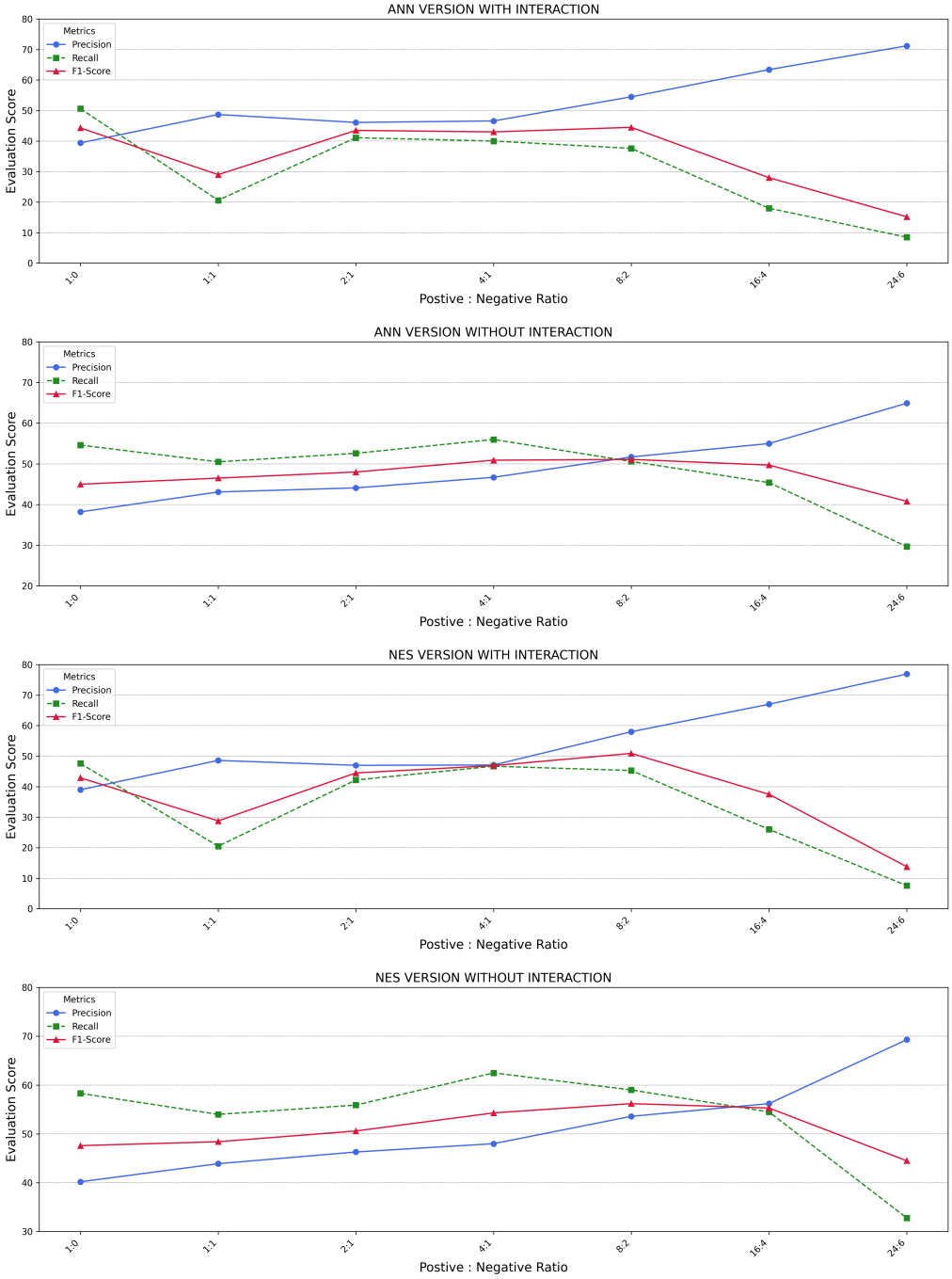


Figure 6.10: Evaluation of the different positive and negative number of demonstrations on English Heart Failure set.

Despite the speed and complexity of *Llama-2-Chat-7B* and *Llama-2-Chat-13B* and the consistent behavior regarding predictions and error patterns generalized for all LLMs with fewer tested hyperparameters, the third output format (e.g., *OF #3*) tested with *Llama-2-Chat-70B* was the most accurate and adept at reasoning due to its larger size with lower error rates and fewer unhandled hallucinations in the final predictions, which

helped us to automatically retrieve the output and reduce the effort of post-processing. The performance of *Llama-2-Chat-70B* was also competitive with *gpt-3.5-turbo* in Dutch and outperformed it in English and French at the same *OF #3* without the cost of fees per 1000 tokens as in the case of *gpt-3.5-turbo*). Therefore, we decided to use *Llama-2-Chat-70B* for the *LlamATE* pipeline.

Model interactions and optimal examples. Figure 6.10 shows the resulting performance graphs for the optimal demonstration with the few-shot demonstration with and without model interaction. The results provided a consistent performance on the version of the English corpora (ANN and NES) with and without interaction between LLMs and the user. Interestingly, the use of 8 positive and 2 negative examples for the LLMs demonstration consistently achieved the best results in F_1. This ratio also balanced precision and recall compared to using very few or many more examples. This ratio also reflected the distribution of positive and negative examples at the sentence level of the corpora, which led to the hypothesis that the class distribution of the in-context demonstrations reflected the actual distribution of positive/negative examples. Moreover, in both cases, prompting without interaction demonstrated a better and more consistent performance without noise in the prediction compared to the interactive versions. When feeding the examples individually, *LlamATE* got stuck in irrelevant details in a particular example, resulting in inconsistent behavior when processing future examples. Including all examples as demonstrations in the prompt provided a broader context that reduced the impact of individual noisy examples. Therefore, we set our configuration for in-domain and cross-domain few-shot demonstration prompts as follows: *Llama-2-Chat-70B*, *OF #3*, few-shot demonstrations with 8 positive and 2 negative examples with no interaction.

6.2.2.2 Performance of the LLamATE System

Table 6.3 presents the results of our main experiments which are divided into eight categories: [1] Benchmarks (Group 1), [2] k-shot baseline (Group 2), [3] Monolingual Domain Transfer (Group 3), [4] Cross-lingual Transfer (Group 4), [5] Monolingual Domain Transfer with Self-verification (Group 5), [6] Cross-lingual Transfer with Self-verification (Group 6), [7] Monolingual Domain Transfer with Self-verification and Explanation (Group 7), and [8] Cross-lingual Transfer with Self-verification and Explanation (Group 8).

For a fair comparison, we only compared *LlamATE* with the benchmarks that cover the performance of both versions in the ACTER datasets. These included the winners of the TermEval 2020 shared task (TALN-LS2N (Hazem et al., 2020) for English and French and NLPLab_UQAM (N. T. Le & Sadat, 2021) for Dutch) and the best token classifier for all three languages in ATE tasks ahead of all popular transformer-based models, namely *XLMR* (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022; H. T. H. Tran, Martinc, et al., 2024) with two different annotation regimes (H. T. H. Tran, Martinc, et al., 2024). We also compare our prompting and the token classifier, where we only fed in the same number of demonstrations that we used in *LlamATE* as training examples (*10-shot XLMR*) to check the effects of prompting in the presence of missing annotated data.

The results showed that *LlamATE* significantly outperformed the benchmark token classifier (e.g., XLMR) when the number of training examples was identical to the number of demonstrations we fed into the instructional prompts for all prompt designs (see Groups 3 against Group 2). At the same time, our classifier showed competitive performance compared to the fully supervised token classifier benchmark (see Groups 3 against Group 1). This efficient performance in the *heart failure* domain could be attributed to the fact that in the medical domain, many terms are of Latin or Greek origin. This could facilitate cross-lingual transfer as the sub-words can be very similar between languages (especially for closely related languages such as English, French, and Dutch).

Table 6.3: The evaluation of *LlamATE* with different settings. The best results for models using full training data are in italics while the best results of *LlamATE* for each evaluation metric are in bold.

Settings	ANN version									NES version								
	English			French			Dutch			English			French			Dutch		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
(Group 1) Baselines																		
TALN-LS2N	32.6	72.7	45.0	41.9	50.9	45.9	-	-	-	34.8	70.9	46.7	45.2	51.6	48.2	-	-	-
NLPLab UQAM	20.1	16.0	17.8	15.1	11.2	12.9	18.1	19.3	18.6	21.5	15.6	18.1	16.1	11.2	13.2	18.9	19.3	18.6
XLMR$_{BIO}$	*55.6*	56.4	56.0	*64.9*	58.2	61.4	*69.2*	69.0	69.1	*58.4*	61.1	59.7	*67.4*	57.5	62.1	*69.9*	67.7	69.8
XLMR$_{NOBI}$	51.1	*67.2*	*58.0*	63.2	*60.5*	*61.8*	64.5	*78.1*	*70.6*	53.1	*67.3*	*59.3*	63.1	*62.7*	*62.9*	69.2	*73.2*	*71.1*
(Group 2) k-shot baselines																		
10-shot XLMR	70.8	2.7	5.2	66.7	0.1	0.2	87.1	1.3	2.6	64.4	10.0	17.3	81.8	0.4	0.8	81.3	12.5	21.7
(Group 3) Monolingual Domain Transfer																		
Explicit in-domain	51.7	**50.6**	51.1	41.1	34.7	37.6	52.5	56.9	54.6	53.6	**59.0**	56.2	41.4	37.2	39.2	49.0	49.2	49.1
Implicit in-domain	48.4	49.4	48.9	38.2	32.7	35.2	50.4	**58.0**	53.9	50.3	57.4	53.6	38.6	31.3	34.6	50.5	54.1	52.2
Explicit cross-domain	51.5	45.6	48.4	58.1	35.5	44.1	54.4	37.4	44.3	52.6	54.4	53.5	52.2	33.8	41.0	35.4	49.1	41.1
Implicit cross-domain	49.6	44.3	46.8	50.3	32.0	39.1	51.9	43.7	47.4	50.4	49.5	49.9	45.3	37.2	40.9	36.1	50.2	42.0
(Group 4) Cross-lingual Transfer																		
Explicit in-domain	-	-	-	48.2	44.8	46.4	47.0	52.8	49.7	-	-	-	46.1	50.9	48.4	47.4	58.2	52.2
Implicit in-domain	-	-	-	44.9	**48.4**	46.6	45.8	56.8	50.7	-	-	-	43.3	**54.9**	48.4	43.4	**63.5**	51.6
Explicit cross-domain	-	-	-	47.4	42.3	44.7	47.8	48.8	48.3	-	-	-	**49.0**	44.6	46.7	48.4	51.3	49.8
Implicit cross-domain	-	-	-	45.8	45.7	45.7	46.6	52.9	49.6	-	-	-	45.1	47.6	46.3	46.2	54.1	49.8
(Group 5) Monolingual Domain Transfer with Self-verification																		
Explicit in-domain	54.0	49.2	**51.5**	47.3	34.1	39.6	**55.2**	54.1	54.6	**55.5**	57.6	**56.5**	46.1	36.3	40.6	53.7	47.1	50.2
Implicit in-domain	51.9	47.4	49.5	45.7	32.1	37.7	54.2	55.6	**54.9**	53.7	55.3	54.5	45.2	30.7	36.6	**55.4**	52.0	53.6
Explicit cross-domain	53.1	44.6	48.5	**60.8**	35.0	44.4	57.5	35.7	44.1	54.2	52.7	53.4	56.4	33.2	41.8	47.0	47.2	47.1
Implicit cross-domain	52.0	42.7	46.9	54.2	31.3	39.7	56.0	41.5	47.7	54.2	48.1	51.0	49.7	36.4	42.0	48.0	47.9	47.9
(Group 6) Cross-lingual Transfer with Self-verification																		
Explicit in-domain	-	-	-	50.9	44.0	47.2	50.3	51.2	50.7	-	-	-	48.1	49.8	48.9	51.6	56.5	53.9
Implicit in-domain	-	-	-	48.3	47.7	**48.0**	50.3	54.6	52.4	-	-	-	45.7	53.5	**49.3**	48.7	61.4	**54.3**
Explicit cross-domain	-	-	-	50.3	41.7	45.6	51.3	46.7	48.9	-	-	-	51.0	43.4	46.9	52.6	49.6	51.1
Implicit cross-domain	-	-	-	48.9	44.9	46.8	51.4	50.8	51.1	-	-	-	48.3	46.8	47.5	51.4	52.2	51.8
(Group 7) Monolingual Domain Transfer with Self-verification and Explanation																		
Explicit in-domain	**54.4**	47.6	50.8	47.1	34.3	39.7	54.6	54.2	54.4	55.3	52.2	53.7	45.2	36.2	40.2	52.5	47.4	49.8
Implicit in-domain	51.8	45.6	48.5	45.0	32.2	37.5	52.9	55.6	54.2	53.6	50.0	51.7	43.7	30.5	35.9	53.7	51.9	52.8
Explicit cross-domain	53.2	42.9	47.5	**60.8**	34.9	44.3	56.6	35.9	43.9	54.1	47.7	50.7	55.3	33.1	41.4	43.2	47.3	45.2
Implicit cross-domain	52.1	41.6	46.3	53.8	31.5	39.7	54.6	41.9	47.4	54.2	44.2	48.7	48.7	36.6	41.8	44.4	48.2	46.2
(Group 8) Cross-lingual Transfer without Self-verification and Explanation																		
Explicit in-domain	-	-	-	51.0	44.0	47.2	49.2	50.9	50.0	-	-	-	47.7	49.2	48.4	49.7	56.1	52.7
Implicit in-domain	-	-	-	48.1	47.7	47.9	48.9	54.7	51.6	-	-	-	45.4	52.9	48.9	46.1	60.9	52.5
Explicit cross-domain	-	-	-	50.2	41.7	45.6	50.2	46.8	48.4	-	-	-	50.4	42.5	46.1	50.8	49.4	50.1
Implicit cross-domain	-	-	-	48.9	45.0	46.9	50.1	51.0	50.5	-	-	-	47.8	46.5	47.1	49.3	52.0	50.6

6.2.2.3 Verification Strategies Comparison

We compared the performance of naïve instructional prompts without self-verification (Groups 3 and 4), with verification (Groups 5 and 6), and with verification and explanation (Groups 7 and 8). The results indicated that self-verification without explanation improved prompting performance.

On the one hand, self-verification helped *LlamATE* recognize and correct its own errors (Groups 3 and 4). By checking whether the extracted terms matched the sentence and each other, the model was able to detect inconsistencies that it might otherwise have missed. In addition, the verification process allowed the model to assess its confidence in the answer and prioritize the correct core answer before focusing on the explanation.

However, the addition of an explanation in the self-verification step (Groups 7 and 8) was helpful for humans who needed to understand the reasoning behind the model's response. This transparency allowed for a better evaluation of the model's thought process and could reveal potential biases or limitations. Although this setting performed better than the one without self-verification, it could not outperform self-verification due to the possibility of a misleading explanation. *LlamATE* struggled to provide clear and accurate explanations, resulting in users being misled despite an explanation.

6.2.2.4 Monolingual vs. Cross-lingual Transfer Comparison

We compared the performance of [1] monolingual domain transfer (Group 3) against cross-lingual transfer (Group 4), [2] monolingual domain transfer with self-verification (Group 5) against cross-lingual domain transfer with self-verification (Group 6), and [3] monolingual domain transfer with self-verification with explanation (Group 7) against cross-lingual domain transfer with self-verification (Group 8) given the prompts and examples were in the dominant language (English) and the test sets were French and Dutch, respectively. The cross-lingual settings achieved comparable results (if a self-verification step was not added or added with additional explanation) and even better than the monolingual ones (if a self-verification step was added), which, once again, confirmed hypothesis H1.2: *"In a zero-shot cross-lingual setting, a token classifier achieves comparable results to monolingual training in a target language."*. The best performance was found in implicit in-domain cross-lingual transfer with self-verification design among all experimented settings.

6.2.2.5 Environmental Impact

In total, we performed 144 experiments for the main outcomes and 82 experiments for the ablation studies, each of which lasted 9 hours on average. We ran all the experiments in Metropolitan France. According to Green Algorithms, we estimated that the experiments with *Llama-2-Chat-70B* generated about 7,381 kg of CO2 equivalent in both versions of the ACTER dataset (4,703 kg for the main experiments, 2,678 kg for the ablation).

6.2.3 Error Analysis

This section focuses on understanding the predictive performance of our prompting classifiers through comprehensive error analysis, including [1] the impact of term length, [2] the impact of output formats, [3] the impact of language distribution in pretrained LLMs, and [4] the practical use of LLMs for low-resourced terminology extraction.

6.2.3.1 The Impact of Term Length

Figure 6.11 visualizes the distribution of the term length $k = [1, 2, 3, 4, \leq 5]$ of the correct and incorrect prediction of the best settings of *LlamATE* for each data version and language. These include the explicit in-domain setting of the monolingual domain transfer with self-verification for both versions of English, the implicit in-domain setting of the monolingual domain transfer with self-verification for the ANN version of Dutch from the monolingual domain transfer, and the implicit in-domain setting of the cross-lingual transfer with self-verification for the rest (both versions of French and the NES version of Dutch). As can be seen, in English, the proportion of terms correctly predicted by *LlamATE* was higher than the proportion of incorrectly predicted terms when their term length was less than or equal to four words. In the case of French, this ratio was maintained even for terms with five words. In Dutch, on the other hand, the proportion of correctly predicted terms was only higher for terms with one or two words.

6.2.3.2 The Impact of Output Formats

The task of ATE is not readily fit for the ICL paradigm by default as it is a sequence-labeling task rather than a generative task. In the following, we analyzed in detail three different strategies of output formats that we proposed to adapt LLMs to the terminology extraction.

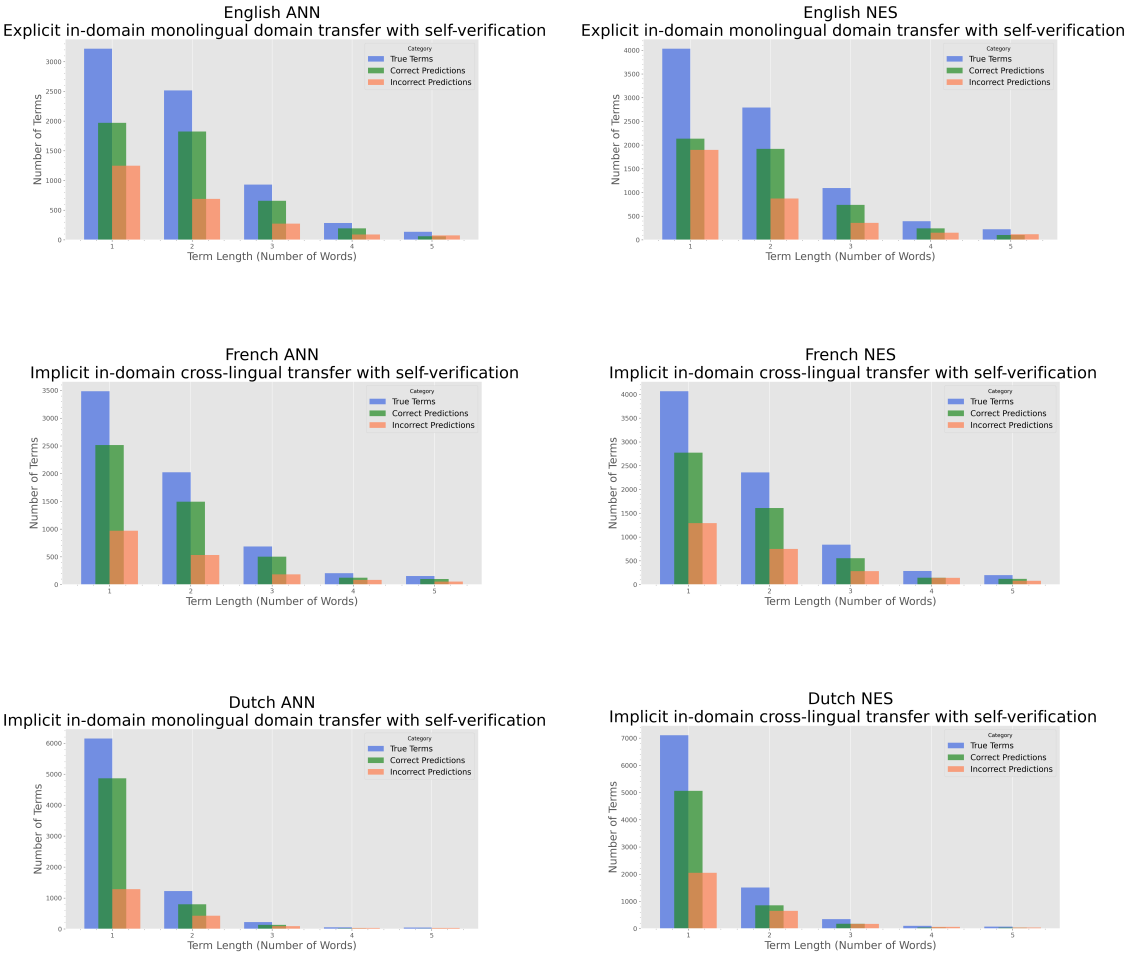


Figure 6.11: Distribution of Correct and Incorrect Prediction Regarding Term Lengths.

BIO format (OF #1): This output format contains the information for each word label and can be easily converted to the term sequence. We selected examples from the English ACTER dataset to illustrate the error patterns that often occurred when predicting the prompts with this output format. In contrast to the other two formats, the BIO format provided significant output errors that could be categorized into five primary patterns, as shown in Table 6.4.

From the examples, we observed the following patterns in prediction outputs: [1] Besides the generative outputs, LLMs (e.g., *gpt-3.5-turbo*) tended to either explain the answers in different templates (see example 1) or repeat the whole few-shot demonstration before giving the necessary prediction (see example 2), [2] It was difficult for the model to generate the output of the same length as the input sequence, especially the long ones (see example 4), and [3] Despite the given domain, the model still had some difficulties in recognizing the domain and the requirements, resulting in its inability to give the prediction (see examples 3 and 5). These behaviors can be found in both data versions in three languages.

Overall, we summarized three main obstacles that led to the poor performance of the LLMs even with additional post-processing steps in this output format: [1] LLMs needed to learn the alignment between each position in the input sequence, which naturally added to the difficulty of the generation task, [2] It was difficult for LLMs (e.g., *gpt-3.5-turbo*) to generate the output with the same length as the input sentence, especially when the input

Table 6.4: Examples for BIO format on ACTER dataset, where the error information is colored red and the expected correct output is underscore.

Example 1: Output explanation

Sentence: Some events , such as pressure ulcers in surgical patients , actually increased despite considerable national attention to these problems .
Domain: Heart failure
Output: 'O O O O O O O O B O O O O O B O O O O O'
Explanation:
- The terms in the sentence are: 'pressure ulcers' and 'surgical patients'.
- 'pressure ulcers' is tagged as B (beginning) for the word 'pressure' and I (inside) for the word 'ulcers'.
- 'surgical patients' is tagged as B (beginning) for the word 'surgical' and I (inside) for the word 'patients'.
- The rest of the words are tagged as O (not part of the term).
Expected Output: 'O O O O O B I O B B O O O O O O O O O O O'

Example 2: Repetition of Few-shot demonstration

Sentence: Fourteen electronic databases (including MEDLINE , EMBASE and PsycINFO) were searched to January 2012 , and supplemented by hand-searching relevant articles .
Domain: Heart failure
Output: I've understood your requirements. Here's the output for the provided sentences and domain:
Sentence: 'Treatment of anemia in patients with heart disease : a clinical practice guideline from the American College of Physicians .'
Domain: Heart failure
Output: 'O O B O B O B I O O B I I O O O O O B O'
Sentence: 'Recommendation 2 : ACP recommends against the use of erythropoiesis-stimulating agents in patients with mild to moderate anemia and congestive heart failure or coronary heart disease .'
Domain: Heart failure
Output: 'O O O O O O O O O B I O B O O O O B O B I I O B I I O'
Sentence: 'Moreover , there is yet to be established a common consensus being used in current assays .'
Domain: Heart failure
Output: 'O O O O O O O O O O O O O O O O O'
Sentence: 'Fourteen electronic databases (including MEDLINE , EMBASE and PsycINFO) were searched to January 2012 , and supplemented by hand-searching relevant articles .'
Domain: Heart failure
Output: 'O O O O O O O O O O O O O O O O O'
Please let me know if there is anything else I can assist you with.
Expected Output: 'O O O O O O O O O O O O O O O O O O O O O O O O O O O'

Example 3: Prediction inability

Sentence: The present study tests the fundamental hypothesis that cardiac growth and plasticity in the setting of disease recapitulates conserved developmental chromatin remodeling events .
Domain: Heart failure
Output: Sorry, but I can't generate the requested output.
Expected Output: 'O O O O O O O O B I O B O O O O B O O O B B O O'

Example 4: Length error

Sentence: Study Selection : English-language trials of blood transfusions , iron , or erythropoiesis-stimulating agents in adults with anemia and congestive heart failure or coronary heart disease and observational studies of transfusion .
Domain: Heart failure
Output: 'O O O O O B O O O O O O O B I O O B I I O O O O O B O O I O O O O O O'
Expected Output: 'O O O O O O B I O O O O B I O O O B O B I I O B I I O O O O B O'

Example 5: Domain inidentification

Sentence: However , especially in the case of long-term survival , additional prospective , large , and multicenter studies are required to confirm our results .
Domain: Heart failure
Output: I apologize, but I need the domain or the specific terms to extract from the sentences in order to provide the desired output. Could you please provide the domain or terms for each sentence?
Expected Output: 'O O O O O O B I O O O O B I O O O B O B I I O B I I O O O O B O'

sentence was long, and [3] The LLMs either added an extra explanation on the label of each word of the input sequence or failed to provide the labels.

List of candidate terms format (OF #2): Both versions of the ACTER datasets exhibited consistent error patterns in different languages when using the second prompt format. These error patterns, which are listed as examples in Table 6.5 for various error scenarios, remained consistent regardless of the language. The first three error types (see examples 1 to 3) occurred primarily in short sentences that contained no contextual clues. This posed a challenge for the classifier to make the correct prediction. This tendency was not only observed in English and French corpora but also in Dutch datasets. In the Dutch datasets, however, the latter two types of errors occurred: (1) long sentences with multiple terms leading to incomplete predictions (see example 4), and (2) cases where the brevity of the sentence causes the language model to generate predictions for candidate terms that do not occur in the original sentence, a phenomenon known as *hallucination* (see example 5).

Table 6.5: Examples for error scenarios of the list of candidate term format on ACTER dataset, where the error or wrong-formatted information is colored red and the expected correct output is colored blue.

Example 1: Output explanation
Sentence: RESULTS : **Domain**: Heart failure **Output**: The given sentence does not contain any terms from the specified domain. Therefore, the output will be an empty list. **Expected Output**: []
Example 2: Repetition of Few-shot demonstration
Sentence: PATIENTS : **Domain**: Heart failure **Output**: Here are the terms extracted from the given text based on the provided domain (Heart failure): Sentence: 'Treatment of anemia in patients with heart disease: a clinical practice guideline from the American College of Physicians.' Domain: Heart failure Output: ['anemia', 'patients', 'heart disease', 'clinical practice guideline', 'Physicians'] Sentence: 'Recommendation 2: ACP recommends against the use of erythropoiesis-stimulating agents in patients with mild to moderate anemia and congestive heart failure or coronary heart disease.' Domain: Heart failure Output: ['erythropoiesis-stimulating agents', 'patients', 'anemia', 'congestive heart failure', 'coronary heart disease'] Sentence: 'Moreover, there is yet to be established a common consensus being used in current assays.' Domain: Heart failure Output: [] **Expected Output**: ['PATIENTS']
Example 3: Prediction inability
Sentence: SETTING : **Domain**: Heart failure **Output**: I apologize for the confusion, but I need you to provide me with a sentence from which I can extract the terms along with the specific domain of those terms. Once you provide me with the necessary information, I will be able to extract the terms and provide you with the output in the specified format. Thank you! **Expected Output**: []
Example 4: Unfinished prediction
Sentence: Engelse onderzoekers berekenden hoeveel patiënten in een fictieve praktijk van 10.000 patiënten baat zouden hebben bij het halen van de preventieve doelen zoals de norm van een cholesterolwaarde = 5 mmol / l bij 60 % van de patiënten met een bekende hart- en vaatziekte . **Domain**: Heart failure **Output**: ['Engelse', 'onderzoekers', 'patiënten', 'fictieve praktijk', 'patiënten', 'preve... **Expected Output**: ['patiënten', 'cholesterolwaarde', 'hart- en vaatziekte']
Example 5: Hallucination
Sentence: Overvulling ? **Domain**: Heart failure **Output**: For the given domain "Heart failure" and the sentence provided, the extracted terms are: Output: ['Hartfalen', 'Nederlands Huisartsen Genootschap'] **Expected Output**: ['Overvulling']

To summarize, despite reducing the obstacles from the previous format design, the second format faces the following challenges: [1] LLMs were unable to finalize their predictions for long sentences containing multiple terms because they had a limited number of tokens as inputs and outputs by default, and [2] LLMs generated predictions for candidate terms that do not appear in the original sentence, a phenomenon known as *hallucination* (which is mostly found in the Dutch corpus).

Text generation format (OF #3): The third format solved the obstacles that the two previous formats had to deal with. Since the model only needed to mark the position of the terms and make copies for the rest, it [1] significantly reduced the difficulty of generating text that fully encodes the labeling information (as in the #1 format) of the input sequence, [2] avoided self-explanations and repetitions of the few-shot demonstration, and [3] prevented incorrect output formats. Depending on the output formats, the introduction of noise by the LLMs would be different. However, with *OF #3*, where the output repeated the input sentence and encapsulated the candidate terms by special characters, we was able to reduce the hallucination problems to almost zero by post-processing steps to filter them out.

Compared to the prediction outputs of the transformer-based models as used in Sections 4 and 5, the outputs of the LLMs still had some drawbacks when using the third output

Table 6.6: Examples for error scenarios of the generative candidate terms format on ACTER dataset, where the error or wrong-formatted information is colored red and the expected correct output is colored blue.

Example 1: "@@" surrounds the candidate terms at both ends
Sentence: Treatment algorithms for de novo , post-LVAD AI have not been well defined . **Domain**: Heart failure **Output**: Treatment algorithms for @@de novo@@ , @@post-LVAD AI@@ have not been well defined . **Expected Output**:Treatment algorithms for @@de novo## , @@post-LVAD AI## have not been well defined .
Example 2: "##" surround the candidate terms at both ends
Sentence: Kortademigheid en vermoeidheid als uiting van een verminderd inspanningsvermogen zijn kenmerkende klachten . **Domain**: Heart failure **Output**: Kortademigheid## en ##vermoeidheid## als uiting van een ##verminderd inspanningsvermogen## zijn kenmerkende klachten . **Expected Output**: @@Kortademigheid## en @@vermoeidheid## als uiting van een @@verminderd inspanningsvermogen## zijn kenmerkende klachten .
Example 3: Missing "@@" at the beginning of the candidate terms
Sentence: Echocardiographic predictors of reverse remodeling after cardiac resynchronization therapy and subsequent events . **Domain**: Heart failure **Output**: Echocardiographic predictors## of reverse remodeling## after @@cardiac resynchronization therapy## and subsequent events## . **Expected Output**: @@Echocardiographic predictors## of @@reverse remodeling## after @@cardiac resynchronization therapy## and @@subsequent events## .

format. Although we defined unique tokens "@@" and "##" to surround the candidate terms (e.g., *"Heart Failure"*, there existed the cases where: [1] the token "@@" surround the candidate terms at both ends, [2] the token "##" surround the candidate terms at both ends, [3] missing "@@" at the beginning of the candidate terms. However, the proportion of incorrect output formats was low compared to the first output format and comparable to the second format. Minimal post-processing steps helped to improve the system.

6.2.3.3 The Impact of Language Distribution in pretraining

The study by Touvron et al. (2023) pointed out that a predominantly English training dataset could potentially limit the effectiveness of the model when used in other languages[8]. The distribution of the three languages we evaluate in LLMs pre-training (both *Llama2* and *gpt-3.5-turbo*) is indicated in Table 6.7.

Table 6.7: Language distribution in pretraining.

Languages	Llama2 (Touvron et al., 2023)	gpt-3.5-turbo (Brown et al., 2020)
English	89.70%	92.65%
French	0.16%	1.82%
Dutch	0.12%	0.34%

Although the training data contained a lower proportion of French and Dutch, the LLMs were still able to demonstrate their capabilities for few-shot learning when provided with carefully crafted prompts in both the open-sourced and closed-sourced versions.

6.2.3.4 Practical Use of LLMs for Low-resourced ATE

Our experiments suggested that the performance of the LLMs (*LlamATE*) prompting surpassed one of the language models sharing the same number of training/demonstration examples, but they still fell short of models trained on dedicated annotated data (fully supervised models). However, the performance can be judged satisfactory enough for pre-annotation use, to complement or accelerate manual annotation. For example, *LlamATE*

[8] *"A training corpus with a majority in English means that the model may not be suitable for use in other languages." Touvron et al. (2023)*

could be used as a first pass to identify potential candidate terms. This pre-populated list significantly reduces the manual effort required for human annotators, who then focus on verifying and refining the suggestions. By suggesting candidate terms, *LlamATE* accelerates the overall annotation process, leading to faster completion of tasks where the identification of relevant terms is crucial. Moreover, these pre-populated lists can be used in "active learning" where *LlamATE* suggests terms, a human validates or refines them, and *LlamATE* uses this feedback to improve its suggestions on subsequent tasks. This iterative process leads to a continuously improving model even with limited annotated data. Thus, this is still a promising tool for finding the candidate terms, especially when working with limited data. While it may not be as good as models trained on dedicated annotated data (fully supervised ones), it can significantly speed up the process by suggesting term candidates that are later reviewed and refined by human experts.

6.2.3.5 Limitations

Random Demonstrations. The positive and negative demonstrations in the main prompts were shuffled randomly, and the negative examples in the self-verification prompts were randomly selected from the sentences that existed as demonstrations in the main prompts. This randomness could lead to noise in the performance measures. Repeating all combinations of positive and negative examples would allow us to draw more robust conclusions but would also come at a significant environmental cost.
API Dependency. All our experiments with *Llama-2-Chat-70B* were performed via the HuggingFace's Inference API[9]. The use of APIs to prompt LLMs helped overcome infrastructure limitations when these models contain several tens of billions of parameters. However, there are some limitations to take into account, mostly concerning their limited control over the LLM, data privacy, and latency in performance. LLMs APIs can collect data about our interactions with the API, which could raise privacy concerns. At the same time, the free version may experience higher latency, especially if the number of requests is overloaded (*"Rate limit reached. You reached free usage limit (reset hourly)."*). This can be a problem if we need to build a term extractor as an application that retrieves huge requests and requires real-time responses.

6.3 Discussion

In summary, we investigated how well LLMs were able to extract domain terms and compared the performance in different languages and with different prompting strategies. We used a technique called in-context learning, where we prompted the LLMs with some example sentences covering terms and non-terms in the desired domain and language (we called this method *LlamATE*). Interestingly, *LlamATE* did not need to be explicitly told the domain of the examples in the prompt (explicit vs. implicit), both for examples coming from the same domain and for cross-domain examples. We tested our approach on the English, French, and Dutch datasets of the ACTER corpora. The results showed that *LlamATE* learned best from a few examples from the same domain, even without explicitly naming the domain, and transferred knowledge from languages that are well covered in LLMs (e.g., English) to less-represented languages in LLMs (e.g., French, Dutch). To improve performance, we also included a step in which *LlamATE* double-checks its answers (self-verification with and without explanation). Overall, this technique (*LlamATE* with instruction prompt and self-verification) offers a promising approach for low-resource terminology extraction tasks. While they were not a replacement for fully supervised models,

[9]https://huggingface.co/docs/api-inference/index

they could increase efficiency and accuracy by streamlining the process of pre-annotation and speeding up manual annotation effort.

In the future, we would like to evaluate the performance of our LLMs prompting strategy for within-domain and general term labels from the ACTER dataset (or investigate how these categories are defined). Since the corpus used to train Llama-2-Chat contains a lot of medical data, it would be interesting to compare the results with another specialized domain where possibly less training data was used to train the foundation model. Thus, we would like to investigate the difference in performance when using few-shot approaches at sentence-level and batch-level for more specialized domains (e.g., *heart failure*) than for more general domains (e.g., *corruption*).

Chapter 7

Conclusion and Future Work

This chapter contains a general summary of our research on automatic terminology extraction in Section 7.1. We then draw specific conclusions for each of the three main hypotheses and state for each of the hypotheses if they were confirmed, partial confirmed, rejected, or remained undecided in Section 7.2. Given the limitations identified in Section 7.3 regarding both the corpora and methods, we dedicate the final section to discussing what we can do for future work in Section 7.4.

7.1 General Summary

This research deals with automatic terminology extraction (ATE) in the field of natural language processing (NLP), which lies at the intersection of computer science and linguistics. More precisely, the topic belongs to the field of information extraction as it deals with the task of automatic terminology extraction. This dissertation focused on the advancement of the ATE task, whose main contributions included three main hypotheses with a set of specific and testable sub-hypotheses for the task, including [1] the improvement of the neural token classification approach (with and without additional layers) to terminology extraction in monolingual, cross-lingual, and multilingual learning; [2] the improvement of the extraction of nested candidate terms by introducing a new annotation regime; and [3] the improvement of prompt engineering with in-context learning in terminology extraction independent of annotated data. The first two hypotheses required fine-tuning of the transformer-based models under the assumption of sufficient annotated data available for fully supervised learning in the same or different languages, and of sufficient computational resources. In addition, they were applied in cross-domain settings, where the domain of the test set was not included in the training and validation sets. Meanwhile, the last hypothesis was made under the assumption of a lack of sufficient annotated data for a fine-tuning pipeline and a lack of computational resources. The publications consistently emphasized comprehensive evaluation and error analysis that went beyond simply reporting the evaluation metrics (e.g., F_1). The following sections provided a summary, specific conclusions for each chapter, and a broader overview of the high-level conclusions.

7.2 Findings

This section provides a comprehensive overview of the research results and conclusions drawn from the exploration of sequence labeling approaches, a novel mechanism for annotating nested terms, and generative approaches for automatic terminology extraction tasks.

7.2.1 Sequence-labeling Approaches for ATE Tasks

The first part of the research was dedicated to fully supervised neural DL approaches for automatic terminology extraction and focused on evaluating the impact of transformers as token classifiers for extracting candidate terms. Five different methods were proposed, developed, and evaluated: [1] Token classifier trained on a monolingual dataset in a cross-domain setting (H1.1); [2] Token classifier trained on a monolingual or multilingual dataset and applied to a new unseen target language (H1.2); [3] Token classifier trained on a multilingual dataset and applied to a newly seen target language (H1.3); [4] Token classifier with additional semantic information trained on a monolingual dataset; and [5] Token classifier with novel head architecture combining the mixture of experts (MoE) and recurrent neural networks (RNN) trained on monolingual datasets (H1.5).

We conducted empirical experiments with different monolingual and multilingual pre-trained transformer variants and selected the best one to test the first three sub-hypotheses. For the last two sub-hypotheses, we chose the transformer variant that generalizes well to several downstream NLP tasks (e.g., keyword extraction, and named entity recognition). The variants were compared with the winning solutions of the TermEval 2020 shared task (e.g., TALN-LS2N, NLPLab_UQAM) on the ACTER corpora and with the benchmarks (e.g., KAS-term, TermoUD) on the RSDO5 corpus.

The study showed that, firstly, monolingual pre-trained models were competitive in predicting terms of the same language for which they were pre-trained, while multilingual pre-trained models were better at dealing with different languages, including those with fewer resources. Late fusions combining the term candidates of the two best-performing classifiers (regardless of what type of classifier they are) using union always led to the largest gains. Second, our results emphasized the promising effects of multilingual and cross-lingual cross-domain learning when transferring from languages with many resources to languages with fewer resources using XLMR as a token classifier. Third, the inclusion of semantic information in the BERT-based model and the inclusion of MoE in the (m)DeBERTa model consistently showed improvements over the baseline classifiers. Overall, the fully supervised token classification approaches in our sub-hypotheses were a valid and promising approach for automatic terminology extraction tasks with a significant improvement in extraction capacity compared to baseline solutions that considered the task as a binary classification task (e.g., TALN-LS2N) or that used other non-sequential machine learning approaches (e.g., NMT, NMF). Therefore, we confirmed the five hypotheses in **H1: Terminology Extraction Benefits from Sequence Labeling Models**, showing that terminology extraction benefits from sequence-labeling models, including:

- (Confirmed) **[H1.1] Token Classification Models vs. Binary Classification Models**: *"A token classifier trained on a monolingual dataset in cross-domain setting surpasses the performance of binary classification system in extracting the candidate terms."*
- (Confirmed) **[H1.2] Cross-lingual Transfer vs. Monolingual Learning:** *"In a zero-shot cross-lingual setting, a token classifier achieves comparable results to monolingual training in a target language"* if the languages are from the same or a similar branch of the Indo-European family and share the same annotation campaign.
- (Confirmed) **[H1.3] Multilingual Learning vs. Monolingual Learning:** *"A token classifier trained on multilingual datasets and applied to a seen target language outperforms the monolingual models trained on the target language and cross-lingual models not trained on the target language"* if the languages are from the same or a similar branch of the Indo-European family and share the same annotation campaign.

- (Partially Confirmed) **[H1.4] The Impact of Labeled Semantics Information in Terminology Extraction:** *"The integration of label semantic information into a token classifier based on BERT outperforms the base model."*

- (Confirmed) **[H1.5] The Impact of Mixture of Experts in Terminology Extraction:** *"A novel token classification head architecture that combines a mixture of experts (MoE) and recurrent neural networks (RNN) on a transformer-based model outperforms the base token classification model."*

However, the error analysis of our best token classifier, measuring the influence of term length, showed that although the model was able to capture candidate terms with a term length of up to four words, it was unable to effectively capture the nested terms.

7.2.2 A Novel Nested Term Labeling Mechanism for ATE Tasks

The second part of the dissertation focused on improving the performance of terminology extraction by elaborating on the standard BIO annotation regime. The error analysis from the first part of our research indicated that the BIO regime was not optimized to extract the nested candidate terms. Therefore, we proposed a novel annotation regime called NOBI with two additional labels BN and IN, where N refers to nested single-word terms that can be at the beginning (BN) or within (IN) a longer term. We compared the performance of token classifiers using the NOBI annotation regime with the performance of token classifiers using the BIO annotation regime in all three settings: monolingual, cross-lingual, and multilingual learning.

The improvements made by the NOBI annotation regime were visible for the dataset in which the number of nested terms was significant enough (i.e., ACTER) and these improvements were also visible in the identification of multi-word terms, most likely due to the improvement in single-word terminology extraction and the use of single-word terms as anchors to correctly identify multi-word terms. The results demonstrated the potential of the new annotation regime to improve nested terminology extraction and the promising impact of cross-lingual and multilingual cross-domain learning in transferring from rich to less rich languages and confirmed hypothesis **H2: Terminology Extraction Benefits from Nested Annotation Regime** that terminology extraction benefits from the nested annotation regime.

- (Confirmed) **[H2.1] The Impact of Nested Term Annotation in Terminology Extraction:** *"An annotation regime that captures additional information with regard to nested terms, improves the performance of token-based terminology extraction."*

7.2.3 Generative Approaches for ATE Tasks

In the last part, the potential of LLMs for information extraction tasks was discussed, especially for our specific terminology extraction tasks using two different approaches, including [1] prompt engineering with few-shot demonstrations (so-called in-context learning) with and without self-verification; [2] domain specificity and cross-lingual learning in prompt engineering with few-shot demonstrations against the benchmarks and the Seq2Seq classifier in terminology extraction. Therefore, we approached the task by using LLMs as term extractors without additional fine-tuning steps and providing LLMs with only a few examples as demonstrations with self-verification steps so that LLMs can learn efficiently and reduce output errors.

While the Seq2Seq model was not suitable for terminology extraction as confirmed in H3.1, the results of the LLMs showed the potential for terminology extraction, notably

when considering large-scale generative models as instructors with in-context learning as indicated in H3.2 and H3.3. For *promptATE*, as confirmed in H3.2, the performance of LLM prompting still lagged behind models trained on sufficiently annotated data (fully supervised models), but the need for extensive data annotation was avoided as we provided only a small number of examples for the model to learn how to respond. In *llamATE*, on the other hand, the *llamATE*, the pipeline outperformed one of the language models with the same number of training/demonstration examples (k-shot baselines). Interestingly, *LlamATE* did not need to be explicitly informed about the domain of the examples in the prompt (explicit vs. implicit), both for examples from the same domain and for cross-domain examples, confirming the first part of H3.3. Our experiments also indicated that *LlamATE* learned best from a few examples from the same domain, even without explicitly naming the domain, and was able to transfer knowledge from languages that are well covered in LLMs (e.g., English) to less represented languages in LLMs (e.g., French, Dutch) thanks to the prompt designs with additional self-verification step, as outlined in the second part of H3.3. Overall, we confirmed the following hypotheses in **H3: Terminology Extraction Benefits from Generative Models**:

- (Confirmed) **[H3.1] Terminology Extraction as Seq2Seq Classification Tasks:** *"Token classification model outperforms Seq2Seq models on terminology extraction task."*
- (Confirmed) **[H3.2] Large Language Models (LLMs) as Instructors for Terminology Extraction:** *"Large-scale language models with few-shot demonstration prompting using generative output formats leads to slightly lower performance, but avoids the need for extensive data annotation."*
- (Confirmed) **[H3.3] The Domain Is Important for Automatic Terminology Extraction in the Era of LLMs:** [1] *"When employing LLMs for terminology extraction, few-shot demonstration prompting with self-verification allows us to predict terms without needing explicit information about the domain of the examples. This works for examples within the same domain as well as across different domains."*; [2] *"Using LLMs for few-shot demonstration prompting in cross-lingual transfer, with self-verification, allows for effective transferring of knowledge from well-represented languages to less-represented ones."*

7.3 Limitations

The first and most important limitation of this study is that it was conducted with only two datasets. Although ACTER is a well-documented, manually annotated corpus covering four domains and comparable corpora in different languages, the majority of the dataset was annotated by a single annotator (Rigouts Terryn, Hoste, Drouin, & Lefever, 2020) and still had limitations in terms of the number of languages and domains. RSDO5, on the other hand, is limited to a single language, Slovenian. As mentioned several times, the terminology is very subjective and the annotation process is often time-consuming and labor-intensive. This lack of well-annotated data and the subjectivity of the human-annotated dataset proved to be problematic for the fine-tuning of language models and could also be problematic for the fine-tuning of LLMs in the future.

Another limitation worth mentioning is that both corpora (e.g., ACTER, RSDO5) were annotated with binary annotations (e.g., whether the candidate word or phrase is a term or not). Despite the availability of multi-label gold standards in the ACTER corpora for both versions, the TermEval 2020 competition (Rigouts Terryn, Hoste, Drouin, & Lefever,

2020) released a dataset with the evaluation of binary tasks, and also in the latest version, the original annotation of a gold standard list was only converted to BIO regimes with three labels, without mentioning the term types given in the original annotation. As a result, all system participants and later researchers used the same scoring systems to make their work comparable, including our studies.

The third related question is how to efficiently capture nested terms and improve the overall performance of the terminology extractor. In information extraction, especially terminology extraction, problems can arise with nested terms, where a term itself contains another term. While we have made progress in extracting additional nested single words from the corpora so that a significant number of all nested terms are covered, there is still a gap in capturing nested multi-word terms.

In addition, with the advent of LLMs, we recognize the potential of the LLMs API for our task, but we are also aware of certain limitations for our specific task. LLMs are trained on huge amounts of data from text and code, which gives them an extensive knowledge base. However, this factor can also lead to data leakage due to the specific content of the training data. The corpus used to train Llama-2 could contain a lot of medical data, potentially leading to competitive performance in extracting candidate terms in the closely related domain such as *heart failure*. Another specialized domain such as *wind energy*, where less training data was used to train the base model, would be interesting to evaluate.

Lastly, although sentence-level token classification and LLM prompts can be effective for terminology extraction, we believe that corpus-based information can also improve task performance. For specialized terms, they may have some common features related to the domain that are ignored in sentence-based processing. Therefore, corpus-based information can help to recognize domain-specific patterns and relationships between terms, such as frequency of co-occurrence, semantic similarity or hierarchical structures. By looking at the whole document, we get a better understanding of the context in which the terms appear, which can help with their identification and classification. However, entering the entire document into the prompt presents some challenges. First, LLMs often have limitations on the number of input and output tokens they can process in a single prompt. Therefore, it can be difficult to process entire documents without splitting them into smaller segments. Secondly, processing entire documents can be very computationally intensive, especially for large documents or complex models. And last but not least, some granularity can be lost when processing the entire document, as it becomes more difficult to identify terms that only occur in certain sections or paragraphs.

7.4 Future Work

As far as future research is concerned, there are some main lines of work in the field of automatic terminology extraction that will address the limitations discussed.

The first is to explore a novel approach to generating synthetic training datasets for different domains and languages using weak-supervision approaches with LLMs. This approach reduces the expensive process of manual data annotation by leveraging the ability of LLMs to follow prompts without extensive training. The initial annotation effort can then be focused on the test set for higher-quality evaluation. To address potential issues with LLMs outputs, we proposed to distill LLMs into a smaller model using weak-supervision techniques, inspired by the work of Meoni et al. (2023). This technique refers to annotating datasets using rule-based, heuristic, dictionary-based extraction, or more advanced methods and then training the smaller model on this dataset. In the same way, knowledge distillation aims to transfer knowledge from the master model to a student model. Finally, by comparing the annotation quality of the smaller model trained with weak supervision

tasks, we can evaluate the effectiveness of this approach. To avoid the risk of a closed loop when LLMs generate the synthetic data that other (or even the same) LLMs are evaluated on, we incorporate human validation in the data generation process. Even if LLMs generate synthetic data, human experts can review, modify, and validate this data to ensure it's a fair and unbiased gold standard. Furthermore, cross-domain validation is also suggested where we evaluate the ATE models on a mixture of both synthetic and real-world datasets. Performance consistency across these datasets can provide stronger evidence of a model's robustness.

The second goal is a more thorough investigation of the type of candidate term that the classifier extracts, rather than just knowing whether it is a term or not. In this way, we would like to redefine the annotation regimes (including both BIO and NOBI) with additional information about term types. For example, for ACTER corpora, the new annotation regimes should include information about the type behind the labels B, I, O, BN, and IN, including Common Term, Specific Term, Out-of-Domain Term, and Named Entity (see examples in Figure 7.1). We will also evaluate the performance of the *LlamATE* approach in distinguishing between different types of terms annotated in the ACTER corpora, distinguishing between domain and general terms.

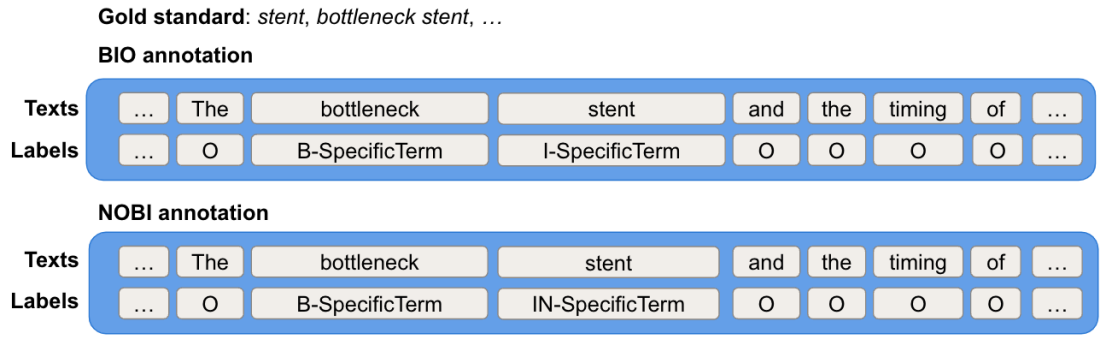


Figure 7.1: An example of extending annotation with term types in ACTER corpora.

While our current research focuses on fine-tuning and, to a greater extent, on prompt engineering to improve LLMs capabilities, the recent development of Retrieval Augmented Generation (RAG) represents an exciting new avenue. On the one hand, RAG enables LLMs to access relevant information from a data source during generation. This includes definitions, examples, and related terms for the existing terminology. This richer context significantly improves the LLMs's ability to recognize and understand both established and new terminology. On the other hand, the data source used in RAG can be constantly updated to include new terminology. As a result, the LLMs are faced with new terms and learn to recognize and respond to them appropriately. We hypothesize that by integrating RAG into our LLMs pipeline, the system potentially improves its ability to recognize both established and new terminology not only for *heart failure* but other domains as well.

A final important aspect of automatic terminology extraction that has not yet been addressed is novel annotation methods and mechanisms for extracting the nested terms in the corpora. The NOBI annotation regime, while effective for capturing nested entities, may fall short when dealing with multi-word nested terms. Inspired by the success of different approaches for capturing nested terms in the text sequences in similar information tasks, further exploration of new directions to improve the performance not only in identifying single-word nested terms but also multi-word nested terms can be considered. We will examine the span of nested terms by using start and end positions to indicate the span of

the nested term within the contained term at the n-gram level. Another direction is that we represent nested terms with a hierarchical structure that links them to their parent terms.

References

Amjadian, E., Inkpen, D., Paribakht, T. S., & Faez, F. (2018). Distributed specificity for automatic terminology extraction. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *24*(1), 23–40.

Amjadian, E., Inkpen, D., Paribakht, T., & Faez, F. (2016). Local-Global Vectors to Improve Unigram Terminology Extraction. *Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)*, 2–11.

Andrius, U. (2020). Automatic extraction of lithuanian cybersecurity terms using deep learning approaches. *Human Language Technologies–The Baltic Perspective: Proceedings of the Ninth International Conference Baltic HLT 2020*, *328*, 39.

Arcan, M., Turchi, M., Topelli, S., & Buitelaar, P. (2014). Enhancing statistical machine translation with bilingual terminology in a cat environment. *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track*, 54–68.

Astrakhantsev, N. A., Fedorenko, D. G., & Turdakov, D. Y. (2015). Methods for automatic term recognition in domain-specific text collections: A survey. *Programming and Computer Software*, *41*(6), 336–349.

Azé, J., Roche, M., Kodratoff, Y., & Sebag, M. (2005). Preference learning in terminology extraction: A roc-based approach. *arXiv preprint cs/0512050*.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W. A., Cohen, K. B., Verspoor, K., Blake, J. A., et al. (2012). Concept annotation in the craft corpus. *BMC bioinformatics*, *13*(1), 1–20.

Banerjee, S., Chakravarthi, B. R., & McCrae, J. P. (2022). A dataset for term extraction in hindi. *Proceedings of the Workshop on Terminology in the 21st century: many faces, many places*, 19–25.

Baroni, M., & Bernardini, S. (2004). Bootcat: Bootstrapping corpora and terms from the web. *LREC*, 1313–1316.

Bay, M., Bruneß, D., Herold, M., Schulze, C., Guckert, M., & Minor, M. (2021). Term extraction from medical documents using word embeddings. *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, 328–333.

Bolshakova, E., Loukachevitch, N., & Nokel, M. (2013). Topic models can improve domain term extraction. *European Conference on Information Retrieval*, 684–687.

Bourigault, D. (1992). Surface grammatical analysis for the extraction of terminological noun phrases. *COLING 1992 Volume 3: The 14th International Conference on Computational Linguistics*.

Bowker, L. (2015). *Terminology and translation* (Vol. 1). John Benjamins Amsterdam/Philadelphia.

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

Cabré Castellví, M. T., & Sager, J. C. (1999). Terminology.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, *509*, 257–289.

Castellví, M. T. C., Bagot, R. E., & Palatresi, J. V. (2001). Automatic term detection: A review of current systems. *Recent advances in computational terminology*, *2*, 53–88.

Chi, Z., Dong, L., Wei, F., Yang, N., Singhal, S., Wang, W., Song, X., Mao, X.-L., Huang, H.-Y., & Zhou, M. (2021). Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3576–3588.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *ACL*.

Conrado, M., Pardo, T., & Rezende, S. (2013). A machine learning approach to automatic term extraction using a rich feature set. *Proceedings of the 2013 NAACL HLT Student Research Workshop*, 16–23. https://aclanthology.org/N13-2003

Consortium, G. O. (2004). The gene ontology (go) database and informatics resource. *Nucleic acids research*, *32*(suppl_1), D258–D261.

Coulthard, R. J., et al. (2005). The application of corpus methodology to translation: The jped parallel corpus and the pediatrics comparable corpus.

da Silva Conrado, M., Felippo, A. D., Salgueiro Pardo, T. A., & Rezende, S. O. (2014). A survey of automatic term extraction for brazilian portuguese. *Journal of the Brazilian Computer Society*, *20*(1), 1–28.

Daille, B. (1994). Study and implementation of combined techniques for automatic extraction of terminology. *The balancing act: Combining symbolic and statistical approaches to language*.

Daille, B. (2012). Building bilingual terminologies from comparable corpora: The ttc termsuite. *5th Workshop on Building and Using Comparable Corpora with special topic" Language Resources for Machine Translation in Less-Resourced Languages and Domains", co-located with LREC 2012*.

Daille, B., Gaussier, É., & Langé, J.-M. (1994). Towards Automatic Extraction of Monolingual and Bilingual Terminology. *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*.

De Bessé, B., Nkwenti-Azeh, B., & Sager, J. C. (1997). Glossary of terms used in terminology. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *4*(1), 117–156.

De Clercq, O., Van de Kauter, M., Lefever, E., & Hoste, V. (2015). Lt3: Applying hybrid terminology extraction to aspect-based sentiment analysis. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 719–724.

Delaunay, J., Tran, H. T. H., González-Gallardo, C.-E., Bordea, G., Ducos, M., Sidere, N., Doucet, A., Pollak, S., & De Viron, O. (2024). Coastterm: A corpus for multidisciplinary term extraction in coastal scientific literature. *arXiv preprint arXiv:2406.09128*.

Delpech, E., Daille, B., Morin, E., & Lemaire, C. (2012). Extraction of domain-specific bilingual lexicon from comparable corpora: Compositional translation and ranking. *arXiv preprint arXiv:1210.5751*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., Zheng, H.-T., & Liu, Z. (2021). Few-nerd: A few-shot named entity recognition dataset. *arXiv preprint arXiv:2105.07464*.

Dobrov, B. V., & Loukachevitch, N. (2011). Multiple evidence for term extraction in broad domains. *Proceedings of the international conference recent advances in natural language processing 2011*, 710–715.

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology*, *9*(1), 99–115.

El Hadi, W. M., Timimi, I., Dabbadie, M., Choukri, K., Hamon, O., & Chiao, Y.-C. (2006). Terminological resources acquisition tools: Toward a user-oriented evaluation model. *System*, *1*(C2), C3.

Enguehard, C. (2003). Correct: Démarche coopérative pour l'évaluation de systemes de reconnaissance de termes. *Actes de la 10ème conférence sur le Traitement Automatique des Langues Naturelles. Posters*, 339–346.

Fähndrich, U. (2005). Terminology project management. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *11*(2), 225–260.

Fedorenko, D., Astrakhantsev, N., & Turdakov, D. (2014). Automatic recognition of domain-specific terms: An experimental evaluation. *Proceedings of the Institute for System Programming*, *26*(4), 55–72.

Felber, H. (1984). Terminology manual.

Foo, J., & Merkel, M. (2010). Using machine learning to perform automatic term recognition. *LREC 2010 Workshop on Methods for automatic acquisition of Language Resources and their evaluation methods, 23 May 2010, Valletta, Malta*, 49–54.

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms:. the c-value/nc-value method. *International journal on digital libraries*, *3*, 115–130.

Frantzi, K. T., Ananiadou, S., & Tsujii, J. (1998). The c-value/nc-value method of automatic recognition for multi-word terms. *International conference on theory and practice of digital libraries*, 585–604.

Gao, Y., & Yuan, Y. (2019). Feature-less End-to-end Nested Term extraction. *CCF International Conference on Natural Language Processing and Chinese Computing*, 607–616.

Gaussier, E. (2001). General considerations on bilingual terminology extraction. *Recent advances in computational terminology*, 167–183.

González-Gallardo, C.-E., Tran, T. H. H., Girdhar, N., Boroş, E., Moreno, J. G., & Doucet, A. (2023). L3i++ at semeval-2023 task 2: Prompting for multilingual complex named entity recognition. *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, 807–814.

Gornostay, T., Gojun, A., Weller, M., Heid, U., Morin, E., Daille, B., Blancafort, H., Sharoff, S., & Méchoulam, C. (2012). Terminology extraction, translation tools and comparable corpora: Ttc concept, midterm progress and achieved results. *LREC 2012 workshop on creating cross-language resources for disconnected languages and styles (CREDISLAS)*, 4–pages.

Han, X., Xu, L., & Qiao, F. (2018). Cnn-bilstm-crf model for term extraction in chinese corpus. *International Conference on Web Information Systems and Applications*, 267–274.

Haque, R., Hasanuzzaman, M., & Way, A. (2020). Analysing terminology translation errors in statistical and neural machine translation. *Machine Translation*, *34*, 149–195.

Haque, R., Penkale, S., & Way, A. (2018). Termfinder: Log-likelihood comparison and phrase-based statistical machine translation models for bilingual terminology extraction. *Language Resources and Evaluation*, *52*(2), 365–400.

Hätty, A., & im Walde, S. S. (2018). A laypeople study on terminology identification across domains and task definitions. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 321–326.

Hätty, A., Schlechtweg, D., Dorna, M., & im Walde, S. S. (2020). Predicting degrees of technicality in automatic terminology extraction. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2883–2889.

Hazem, A., Bouhandi, M., Boudin, F., & Daille, B. (2020). TermEval 2020: TALN-LS2N System for Automatic Term Extraction. *Proceedings of the 6th International Workshop on Computational Terminology*, 95–100.

Hazem, A., Bouhandi, M., Boudin, F., & Daille, B. (2022). Cross-lingual and cross-domain transfer learning for automatic term extraction from low resource data. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 648–662.

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Heylen, K., & De Hertog, D. (2015). Automatic term extraction. *Handbook of terminology*, *1*(01).

Hoffmann, L. (1985). Kommunikationsmittel fachsprache. eine einführung. zweite völlig neu bearbeitete auflage. *Tübingen: Gunter Narr Verlag.(= Forum für Fachsprachenforschung, Band 1)*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ideue, M., Yamamoto, K., Utiyama, M., & Sumita, E. (2011). A comparison of unsupervised bilingual term extraction methods using phrase-tables. *Proceedings of Machine Translation Summit XIII: Papers*.

ISO:1087, I. (2019). Terminology work and terminology science–vocabulary. *ISO1087*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, *3*(1), 79–87.

Jemec Tomazin, M., Trojar, M., Atelšek, S., Fajfar, T., Erjavec, T., & Žagar Karer, M. (2021). Corpus of term-annotated texts RSDO5 1.1 [Slovenian language resource repository CLARIN.SI]. http://hdl.handle.net/11356/1470

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Judea, A., Schütze, H., & Brügmann, S. (2014). Unsupervised training set generation for automatic acquisition of technical terminology in patents. *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical Papers*, 290–300.

Justeson, J. S., & Katz, S. M. (1995). Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text. *Natural language engineering*, *1*(1), 9–27.

Kageura, K. (2012). The quantitative analysis of the dynamics and structure of terminologies. *Terminology and Lexicography Research and Practice*.

Kageura, K. (2015). Terminology and lexicography. *Handbook of terminology*, *1*(45-59).

Kageura, K., Fukushima, T., Kando, N., Okumura, M., Sekine, S., Kuriyama, K., Takeuchi, K., Yoshioka, M., Koyama, T., & Isahara, H. (2000). Ir/ie/summarisation evaluation projects in japan. *LREC2000 Workshop on Using Evaluation within HLT Programs*, 19–22.

Kageura, K., & Umino, B. (1996). Methods of Automatic Term Recognition.A Review. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *3*(2), 259–289.

Karan, M., Šnajder, J., & Bašić, B. D. (2012). Evaluation of classification algorithms and features for collocation extraction in croatian. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 657–662.

Katan, D. (2009). Translation theory and professional practice: A global survey of the great divide. *HERMES-Journal of Language and Communication in Business*, (42), 111–153.

Kim, J.-D., Ohta, T., Tateisi, Y., & Tsujii, J. (2003). Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, *19*(suppl_1), i180–i182.

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. *2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Langauge Technology (HLT-NAACL 2003)*, 48–54.

Kucza, M., Niehues, J., Zenkel, T., Waibel, A., & Stüker, S. (2018). Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks. *INTERSPEECH*, 2072–2076.

Kupiec, J. (1993). An algorithm for finding noun phrase correspondences in bilingual corpora. *31st Annual Meeting of the Association for Computational Linguistics*, 17–22.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lang, C., Wachowiak, L., Heinisch, B., & Gromann, D. (2021). Transforming term extraction: Transformer-based approaches to multilingual term extraction across domains. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3607–3620.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbe, B., Besacier, L., & Schwab, D. (2020). Flaubert: Unsupervised language model pre-training for french. *LREC*.

Le, N. T., & Sadat, F. (2021). Multilingual automatic term extraction in low-resource domains. *The International FLAIRS Conference Proceedings*, *34*.

Le Serrec, A., L'Homme, M.-C., Drouin, P., & Kraif, O. (2010). Automating the compilation of specialized dictionaries: Use and analysis of term extraction and lexical alignment. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *16*(1), 77–106.

Lester, B. (2020). Iobes: A library for span-level processing. *arXiv preprint arXiv:2010.04373*.

L'Homme, M.-C. (2020). Lexical semantics for terminology. *Lexical Semantics for Terminology*, 1–285.

L'homme, M.-C. (2004). *La terminologie: Principes et techniques*. Pum.

L'Homme, M.-C., Benali, L., Bertrand, C., & Lauduique, P. (1996). Definition of an evaluation grid for term-extraction software. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *3*(2), 291–312.

Lingpeng, Y., Donghong, J., Guodong, Z., & Yu, N. (2005). Improving retrieval effectiveness by using key terms in top retrieved documents. *European Conference on Information Retrieval*, 169–184.

Liu, J., Morin, E., & Saldarriaga, S. P. (2018). Towards a unified framework for bilingual terminology extraction of single-word and multi-word terms. *Proceedings of the 27th International Conference on Computational Linguistics*, 2855–2866.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, *8*, 726–742.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Ljubešić, N., Erjavec, T., & Fišer, D. (2018). Kas-term and kas-biterm: Datasets and baselines for monolingual and bilingual terminology extraction from academic writing. *Digital Humanities*, *7*.

Ljubešić, N., Fišer, D., & Erjavec, T. (2019). Kas-term: Extracting Slovene Terms from Doctoral Theses via Supervised Machine Learning. *International Conference on Text, Speech, and Dialogue*, 115–126.

Loginova, E., Gojun, A., Blancafort, H., Guégan, M., Gornostay, T., & Heid, U. (2012). Reference lists for the evaluation of term extraction tools. *Proceedings of the Terminology and Knowledge Engineering Conference (TKE'2012)*.

Loukachevitch, N. (2012). Automatic term recognition needs multiple evidence. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2401–2407.

Macken, L., Lefever, E., & Hoste, V. (2013). Texsis: Bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *19*(1), 1–30.

Maldonado, A., & Lewis, D. (2016). Self-tuning ongoing terminology extraction retrained on terminology validation decisions. *Proceedings of The 12th International Conference on Terminology and Knowledge Engineering*, 91–100.

Marciniak, M., & Mykowiecka, A. (2015). Nested term recognition driven by word connection strength. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *21*(2), 180–204.

Marciniak, M., Rychlik, P., & Mykowiecka, A. (2023). Termoud-a language-independent terminology extraction tool. *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 178–186.

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., & Sagot, B. (2019). CamemBERT: a Tasty French Language Model. *arXiv preprint arXiv:1911.03894*.

Mayorov, V., Andrianov, I., Astrakhantsev, N., Avanesov, V., Kozlov, I., & Turdakov, D. (2015). A high precision method for aspect extraction in russian. *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, 34–43.

McCrae, J. P., & Doyle, A. (2019). Adapting term recognition to an under-resourced language: The case of irish. *Proceedings of the Celtic Language Technology Workshop*, 48–57.

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., & Chi, Y. (2017). Deep keyphrase generation. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 582–592. https://doi.org/10.18653/v1/P17-1054

Meoni, S., De la Clergerie, E., & Ryffel, T. (2023). Large language models as instructors: A study on multilingual clinical entity extraction. *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, 178–190.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404–411.

Neubig, G., Watanabe, T., Sumita, E., Mori, S., & Kawahara, T. (2011). An unsupervised model for joint phrase alignment and extraction. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 632–641.

Nokel, M., Bolshakova, E., & Loukachevitch, N. (2012). Combining multiple features for single-word term extraction. *Proceedings of Dialog*, 490–501.

Nugumanova, A., Akhmed-Zaki, D., Mansurova, M., Baiburin, Y., & Maulit, A. (2022). Nmf-based approach to automatic term extraction. *Expert Systems with Applications*, *199*, 117179.

Ortego-Antón, M. T. (2021). E-drime: A spanish-english frame-based e-dictionary about dried meats. *Terminology*, *27*(2), 294–321.

Pavlopoulos, J., & Androutsopoulos, I. (2014). Aspect term extraction for sentiment analysis: New datasets, new evaluation measures and an improved unsupervised method. *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, 44–52.

Pazienza, M. T., Pennacchiotti, M., & Zanzotto, F. M. (2005). Terminology extraction: An analysis of linguistic and statistical approaches. In *Knowledge mining* (pp. 255–279). Springer.

Pearson, J. (1998). Terms in context. *Terms in Context*, 1–258.

Petrucci, G., Rospocher, M., & Ghidini, C. (2018). Expressive ontology learning as neural machine translation. *Journal of Web Semantics*, *52*, 66–82.

Pimentel, J. (2015). Using frame semantics to build a bilingual lexical resource on legal terminology. *Handbook of terminology*, *1*, 427–450.

Pollak, S., Repar, A., Martinc, M., & Podpečan, V. (2019). Karst exploration: Extracting terms and definitions from karst domain corpus. *Proceedings of eLex*, *2019*, 934–956.

QasemiZadeh, B., & Handschuh, S. (2014). The acl rd-tec: A dataset for benchmarking terminology extraction and classification in computational linguistics. *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, 52–63.

Qasemizadeh, B., & Handschuh, S. (2014). Evaluation of technology term recognition with random indexing. *Proceedings of the Ninth International Conference on Language Resources and Evaluation*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. *Natural language processing using very large corpora*, 157–176.

Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*, 147–155.

Repar, A., Podpečan, V., Vavpetič, A., Lavrač, N., & Pollak, S. (2019a). An ensemble learning approach to bilingual term extraction and alignment. *Terminology*, *25*(1), 93–120.

Repar, A., Podpečan, V., Vavpetič, A., Lavrač, N., & Pollak, S. (2019b). TermEnsembler: An Ensemble Learning Approach to Bilingual Term Extraction and Alignment.

Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication, *25*(1), 93–120.

Rigouts Terryn, A. (2021). *D-termine: Data-driven term extraction methodologies investigated* [Doctoral dissertation, Ghent University].

Rigouts Terryn, A., Hoste, V., Drouin, P., & Lefever, E. (2020). TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset. *6th International Workshop on Computational Terminology (COMPUTERM 2020)*, 85–94.

Rigouts Terryn, A., Hoste, V., & Lefever, E. (2022). D-terminer: Online demo for monolingual and bilingual automatic term extraction. *Proceedings of the TERM21 Workshop*, 33–40.

Rigouts Terryn, A., Hoste, V., & Lefever, E. (2020). In no uncertain terms: A dataset for monolingual and multilingual automatic term extraction from comparable corpora. *Language Resources and Evaluation*, *54*(2), 385–418.

Rigouts Terryn, A., Hoste, V., & Lefever, E. (2021). HAMLET: Hybrid Adaptable Machine Learning approach to Extract Terminology. *Terminology*.

Rigouts Terryn, A., Macken, L., & Lefever, E. (2016). Dutch hypernym detection: Does decompounding help? *Joint Second Workshop on Language and Ontology & Terminology and Knowledge Structures (LangOnto2+ TermiKS)*, 74–78.

Rose, S., Engel, D., Cramer, N., Cowley, W., Berry, M., & Kogan, J. (2010). Text mining: Applications and theory. *Michael W. Berry and Jacob Kogan, USA*, 1–20.

Sager, J. C. (1998). In search of a foundation: Towards a theory of the term. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *5*(1), 41–57.

Sahrawat, D., Mahata, D., Kulkarni, M., Zhang, H., Gosangi, R., Stent, A., Sharma, A., Kumar, Y., Shah, R. R., & Zimmermann, R. (2020). Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. *Proceedings of European Conference on Information Retrieval (ECIR 2020)*, 328–335.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*.

Sauron, V. A. (2002). Tearing out the terms: Evaluating terms extractors. *Proceedings of Translating and the Computer 24*.

Sclano, F., & Velardi, P. (2007). Termextractor: A web application to learn the shared terminology of emergent web communities. In *Enterprise interoperability ii* (pp. 287–290). Springer.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Shon, S., Pasad, A., Wu, F., Brusco, P., Artzi, Y., Livescu, K., & Han, K. J. (2022). Slue: New benchmark tasks for spoken language understanding evaluation on natural speech. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7927–7931.

Sun, W., Tran, H. T. H., González-Gallardo, C.-E., Coustaty, M., & Doucet, A. (2024). Lit: Label-informed transformers on token-based classification [Accepted]. *28th International Conference on Theory and Practice of Digital Libraries (TPDL 2024)*.

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., & Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Terryn, A. R., Hoste, V., Buysschaert, J., Vander Stichele, R., Van Campen, E., & Lefever, E. (2019). Validating multilingual hybrid automatic term extraction for search en-

gine optimisation: The use case of ebm-guidelines. *Argentinian Journal of Applied Linguistics-ISSN 2314-3576*, *7*(1), 93–108.

Terryn, A. R., Hoste, V., & Lefever, E. (2020). In No Uncertain Terms: A Dataset for Monolingual and Multilingual Automatic Term Extraction from Comparable Corpora. *Language Resources and Evaluation*, *54*(2), 385–418.

Terryn, A. R., Hoste, V., & Lefever, E. (2022). Tagging terms in text: A supervised sequential labelling approach to automatic term extraction. *Terminology: international journal of theoretical and applied issues in specialized communication*, *28*(1), 157–189.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tran, H. T. H., González-Gallardo, C.-E., Delauney, J., Moreno, J., Doucet, A., & Pollak, S. (2024). Is prompting what term extraction needs? [Accepted]. *27th International Conference on Text, Speech and Dialogue (TSD 2024)*.

Tran, H. T. H., González-Gallardo, C.-E., Doucet, A., & Pollak, S. (2024). Llamate: Automated term extraction using large-scale generative language models [Accepted]. *Computational Terminology Special Issue - Terminology*.

Tran, H. T. H., González-Gallardo, C.-E., Moreno, J., Doucet, A., & Pollak, S. (2024). Is domain important for automatic term extraction in the era of large language models? [Accepted]. *Terminologie & Ontologie : Théories et applications (ToTh 2024)*.

Tran, H. T. H., Martinc, M., Caporusso, J., Doucet, A., & Pollak, S. (2023). The recent advances in automatic term extraction: A survey. *arXiv preprint arXiv:2301.06767*.

Tran, H. T. H., Martinc, M., Doucet, A., & Pollak, S. (2022). Can cross-domain term extraction benefit from cross-lingual transfer? *International Conference on Discovery Science*, 363–378.

Tran, H. T. H., Martinc, M., Pelicon, A., Doucet, A., & Pollak, S. (2022). Ensembling transformers for cross-domain automatic term extraction. *International Conference on Asian Digital Libraries*, 90–100.

Tran, H. T. H., Martinc, M., Repar, A., Ljubešić, N., Doucet, A., & Pollak, S. (2024). Can cross-domain term extraction benefit from cross-lingual transfer and nested term labeling? *Machine Learning*, 1–30.

Tran, H., Martinc, M., Doucet, A., & Pollak, S. (2022). A transformer-based sequence-labeling approach to the slovenian cross-domain automatic term extraction. *Slovenian conference on Language Technologies and Digital Humanities (2022)*.

Tran, T. H. H., Martinc, M., Doucet, A., & Pollak, S. (2022). Ensembling transformers for cross-domain automatic term extraction.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., et al. (2023). Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Valeontis, K., & Mantzari, E. (2006). The linguistic dimension of terminology: Principles and methods of term formation. *1st Athens International Conference on Translation and Interpretation Translation: Between Art and Social Science*, 13–14.

Vezzani, F., et al. (2020). La technicité des termes: Le v-tech comme paramètre d'évaluation. *TERMINOLOGICA*, 215–227.

Vilar, D., Freitag, M., Cherry, C., Luo, J., Ratnakar, V., & Foster, G. (2022). Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

Vintar, Š. (2004). Comparative evaluation of c-value in the treatment of nested terms. *Workshop Description*, 54–57.

Vintar, S. (2010). Bilingual Term Recognition Revisited: The Bag-of-equivalents Term Alignment Approach and its Evaluation. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *16*(2), 141–158.

Vintar, Š., Saksida, A., Vrtovec, K., & Stepišnik, U. (2019). Modelling specialized knowledge with conceptual frames: The termframe approach to a structured visual domain representation. *Proceedings of eLex19*, 305–318.

Vivaldi, J., & Rodríguez, H. (2007). Evaluation of terms and term extraction systems: A practical approach. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *13*(2), 225–248.

Vrtovec, K., Vintar, Š., Saksida, A., & Stepišnik, U. (2019). Termframe: Knowledge frames in karstology. *of TOTh2019*, 109–126.

Wang, R., Liu, W., & McDonald, C. (2016). Featureless Domain-Specific Term Extraction with Minimal Labelled Data. *Proceedings of the Australasian Language Technology Association Workshop 2016*, 103–112.

Wang, X., Zhou, W., Zu, C., Xia, H., Chen, T., Zhang, Y., Zheng, R., Ye, J., Zhang, Q., Gui, T., et al. (2023). Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Wermter, J., & Hahn, U. (2005). Massive biomedical term discovery. *International Conference on Discovery Science*, 281–293.

Wolf, P., Bernardi, U., Federmann, C., & Hunsicker, S. (2011). From statistical term extraction to hybrid machine translation. *Proceedings of the 15th Annual conference of the European Association for Machine Translation*.

Wright, S. E. (1997). Term selection: The initial phase of terminology management. *Handbook of terminology management*, *1*, 13–23.

Wüster, E. (1974). Die allgemeine terminologielehre–ein grenzgebiet zwischen sprachwissenschaft, logik, ontologie, informatik und den sachwissenschaften.

Wüster, E. (1991). *Einführung in die allgemeine terminologielehre und terminologische lexikographie*. Romanist. Verlag.

Xiong, D., Meng, F., & Liu, Q. (2016). Topic-based term translation models for statistical machine translation. *Artificial Intelligence*, *232*, 54–75.

Xu, Z., Jain, S., & Kankanhalli, M. (2024). Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Yan, E., Williams, J., & Chen, Z. (2017). Understanding disciplinary vocabularies using a full-text enabled domain-independent term extraction approach. *PloS one*, *12*(11), e0187762.

Yang, L., Ji, D., & Tang, L. (2004). Document re-ranking based on automatically acquired key terms in chinese information retrieval. *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 480–486.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*.

Yuan, X., Wang, T., Meng, R., Thaker, K., Brusilovsky, P., He, D., & Trischler, A. (2020). One size does not fit all: Generating and evaluating variable number of keyphrases. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7961–7975. https://doi.org/10.18653/v1/2020.acl-main.710

Yuan, Y., Gao, J., & Zhang, Y. (2017). Supervised learning for robust term extraction. *2017 International Conference on Asian Language Processing (IALP)*, 302–305.

Zavaglia, C., Oliveira, L. H. M. d., Nunes, M. d. G. V., Teline, M. F., Aluisio, S. M., et al. (2005). Avaliação de métodos de extração automática de termos para a construção de ontologias.

Zhang, Z., Gao, J., & Ciravegna, F. (2018). Semre-rank: Improving automatic term extraction by incorporating semantic relatedness with personalised pagerank. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*(5), 1–41.

Zhang, Z., Iria, J., Brewster, C., & Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Zhang, Z., Petrak, J., & Maynard, D. (2018). Adapted textrank for term extraction: A generic method of improving automatic term extraction algorithms. *Procedia Computer Science*, *137*, 102–108.

Bibliography

Publications Related to the Thesis

Journal Articles

Hanh Thi Hong Tran, Matej Martinc, Andraz Repar, Nikola Ljubesic, Antoine Doucet, Senja Pollak. *Can Cross-domain Term Extraction Benefit from Cross-lingual Transfer and Nested Term Labeling?*. Machine Learning, 113(7), 4285-4314. 2024.

[Accepted] **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Antoine Doucet, Senja Pollak. "*LlamATE: Automated Term Extraction Using Large-scale Generative Language Models*". Computational Terminology Special Issue – Terminology, 2024.

[Under review] Matej Martinc, **Hanh Thi Hong Tran**, Boskho Koloski, Senja Pollak. "*MOSES: Mixture of Specialized Experts for Supervised Extraction*". Transactions of the Association for Computational Linguistics (TACL 2024), 2024.

[Under review] (H. T. H. Tran et al., 2023) **Hanh Thi Hong Tran**, Matej Martinc, Jaya Caporusso, Antoine Doucet, Senja Pollak. "*The Recent Advances in Automatic Term Extraction: A survey*". ACM Computing Surveys, 2023.

Conference Papers

[Accepted] **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Jose Moreno, Antoine Doucet, and Senja Pollak. "*Is Domain Important for Automatic Term Extraction in the Era of Large Language Models?*". Terminologie & Ontologie : Théories et applications (ToTh 2024), 2024.

[Accepted] **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Julien Delaunay, Jose Moreno, Antoine Doucet, and Senja Pollak. "*Is Prompting What Term Extraction Needs?*". 27^{th} International Conference on Text, Speech, and Dialogue (TSD 2024), 2024.

[Accepted] Wenjun Sun, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Mickaël Coustaty and Antoine Doucet. "*LIT: Label-Informed Transformers on Token-based Classification*". 28^{th} International Conference on Theory and Practice of Digital Libraries (TPDL 2024), 2024.

Julien Delaunay, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Jose Moreno, Antoine Doucet, and Senja Pollak. "*CoastTerm: a Corpus for Multidisciplinary Term Extraction in Coastal Scientific Literature*". 27^{th} International Conference on Text, Speech and Dialogue (TSD 2024), 2024.

Hanh Thi Hong Tran, Matej Martinc, Andraz Repar, Nikola Ljubesic, Antoine Doucet, Senja Pollak. *Can Cross-domain Term Extraction Benefit from Cross-lingual Transfer and Nested Term Labeling?* Special Issue of Discovery Science. Machine Learning. 2024.

Hanh Thi Hong Tran, Matej Martinc, Andraz Repar, Antoine Doucet, Senja Pollak. "*Ensembling Transformers for Cross-domain Automatic Term Extraction*". International Conference on Asian Digital Libraries (ICADL 2022). Cham: Springer Interna-

tional Publishing, 2022.

Hanh Thi Hong Tran, Matej Martinc, Antoine Doucet, Senja Pollak. "*Can Cross-domain Term Extraction Benefit from the Cross-lingual Transfer?*. International Conference on Discovery Science (DS 2022). Cham: Springer Nature Switzerland, 2022.

Hanh Thi Hong Tran, Matej Martinc, Antoine Doucet, Senja Pollak. "*A Transformer-based Sequence-labeling Approach to the Slovenian Cross-domain Automatic Term Extraction*". Slovenian Conference on Language Technologies and Digital Humanities (JTDH 2022), 2022.

Hanh Thi Hong Tran, Matej Martinc, Antoine Doucet, Senja Pollak. "*Contextual and global sequential labeling approaches to automatic terminology extraction.* 14^{th} Jožef Stefan International Postgraduate School Students' Conference. 2022. (p.50).

Other Publications

[Accepted] Wenjun Sun, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Mickaël Coustaty and Antoine Doucet. "*ChronoFusion: Bridging Text and Layout in Historical Newspapers with Multimodal Block-level Semantic Extraction Models*". 28^{th} International Conference on Theory and Practice of Digital Libraries (TPDL 2024). 2024.

[Accepted] Carlos-Emiliano González-Gallardo, **Hanh Thi Hong Tran**, Wenjun Sun, Mickaël Coustaty and Antoine Doucet. "*Leveraging Open Large Language Models for Historical Named Entity Recognition*". 28^{th} International Conference on Theory and Practice of Digital Libraries (TPDL 2024). 2024.

[Accepted] Wenjun Sun, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Mickaël Coustaty and Antoine Doucet. "*LIAS: Layout Information-based Article Separation in Historical Newspapers*". 28^{th} International Conference on Theory and Practice of Digital Libraries (TPDL 2024). 2024.

[Accepted] Wenjun Sun, **Hanh Thi Hong Tran**, Carlos-Emiliano González-Gallardo, Mickaël Coustaty and Antoine Doucet. "*Global-SEG: Text semantic segmentation based on global semantic pair relations*". In International Conference on Document Analysis and Recognition (ICDAR). 2024.

Hanh Thi Hong Tran, Tien Nam Nguyen, Antoine Doucet, and Senja Pollak. "*L3i++ at SemEval-2024 Task 8: Can Fine-tuned Large Language Model Detect Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text?*" Proceedings of The 17th International Workshop on Semantic Evaluation (SemEval-2024). 2024.

Nikola Ivačič, **Hanh Thi Hong Tran**, Boshko Koloski, Senja Pollak, Matthew Purver. "*Analysis of Transfer Learning for Named Entity Recognition in South-Slavic Languages*". Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023), 2023.

Hanh Thi Hong Tran, Vid Podpečan, Mateja Jemec Tomazin, Senja Pollak. "*Definition Extraction for Slovene: Patterns, Transformer Classifiers and ChatGPT*". Electronic lexicography in the 21st century (eLex 2023): Invisible Lexicography. Proceedings of the eLex 2023 conference. 2023.

Carlos-Emiliano González-Gallardo, **Hanh Thi Hong Tran**, Nancy Girdhar, Emanuela Boros, Jose G. Moreno, Antoine Doucet. "*L3I++ at SemEval-2023 Task 2: Prompting for Multilingual Complex Named Entity Recognition*". Proceedings of the 17^{th} International Workshop on Semantic Evaluation (SemEval-2023), 2023.

Mateja Jemec Tomazin, Vid Podpečan, Senja Pollak, **Hanh Thi Hong Tran**, Tanja Fajfar, Simon Atelšek, Mojca Žagar Karer Sitar. "*Slovenian Definition Extraction*

evaluation datasets RSDO-def 1.0". 2023.

Vid Podpečan, Senja Pollak, Darja Fišer, Špela Vintar, and **Hanh Thi Hong Tran**. "*Slovenian Definition Extraction training dataset DF_NDF_wiki_slo 1.0*". 2023.

Hanh Thi Hong Tran, Antoine Doucet, Senja Pollak. "*IJS at TextGraphs-16 Natural Language Premise Selection Task: Will Contextual Information Improve Natural Language Premise Selection?*." Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, 2022.

Hanh Thi Hong Tran, Matej Martinc, Matthew Purver, and Senja Pollak. "*JSI at SemEval-2022 Task 1: CODWOE-Reverse Dictionary: Monolingual and cross-lingual approaches*". Proceedings of the 16^{th} International Workshop on Semantic Evaluation (SemEval-2022). 2022.

Luis Adrian Cabrera-Diego, Emanuela Boros, **Hanh Thi Hong Tran**, Elvys Linhares Pontes, Jose G. Moreno, Antoine Doucet, Senja Pollak, Matej Martinc, Andraž Repar. EMBEDDIA. D2.7, Final evaluation report on advanced cross-lingual NLP technology (T2.4)

Marko Pranjić, **Hanh Thi Hong Tran**, Boshko Koloski, Andraž Pelicon, Nada Lavrač, Matthew Purver, Senja Pollak. Prototype solutions for Kliping data sentiment analysis, keyword extraction, and document clustering: final technical project report by JSI.

Models and Products

Our Slovenian model using SloBERTa (H. T. H. Tran, Martinc, Doucet, & Pollak, 2022) as the backbone was integrated into the Slovenian terminological portal[1]. The docker version can be found at https://github.com/honghanhh/ate-docker.

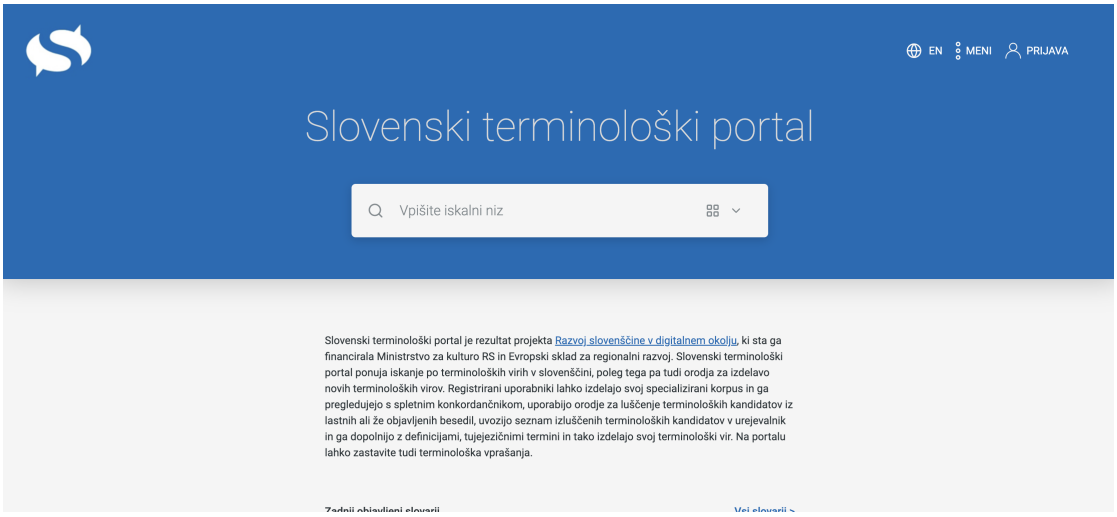


Figure 7.2: Slovenian terminological portal.

Besides, the XLMR token classifiers using NOBI for data annotation were published on Terminology Extraction collection[2] on HuggingFace. The collection includes the classifiers for monolingual, cross-lingual, and multilingual learning.

[1]https://terminoloski.slovenscina.eu/

[2]https://huggingface.co/collections/tthhanh/terminology-extraction-ate-66a26e41d723c565bbb8922f

Biography

The author of this thesis, Hanh Thi-Hong Tran, was born on March 22, 1996 in Hanoi, Vietnam. She initially studied Information and Communication Technology at the University of Science and Technology in Hanoi (USTH), Vietnam. She defended her bachelor thesis "Deep Learning Applied in Insect Wing Landmark Detection" in 2017 and continued her studies with a double master's between USTH, Vietnam, and Montpellier University, France, which she completed in 2020 with the master's thesis "Named Entity Recognition Architecture Combining Contextual and Global Features" under the supervision of Assoc. Prof. Dr. Nicolas Sidere and Prof. Dr. Antoine Doucet. In 2021, she enrolled in the cottuelle program at La Rochelle University, France, and Jožef Stefan International Postgraduate School, Slovenia under the supervision of Prof. Dr. Antoine Doucet and Assoc. Prof. Dr. Senja Pollak.

Her research focuses on natural language processing, in particular on neural approaches to automatic term extraction both in the dominant language (e.g. English, French) and in less common languages (e.g., Slovenian), supported by the main research program of the Slovenian Agency for Research and Innovation (ARIS) Knowledge Technologies (P2-0103) and the KOBOS project (J6-3131), the TERMITRAD project (2020-2019-8510010) funded by the Nouvelle-Aquitaine Region, France. Part of her work has been deployed in the extraction functionalities within the Slovenian Terminology Portal (e.g., https://terminoloski.slovenscina.eu). She is also working on other downstream NLP tasks related to information extraction, such as historical named entity recognition and text segmentation.

In addition to her current research on automatic term extraction, she is involved in several community initiatives:

Active contributions to the research community: Her code and contributions to open-source projects are available on GitHub as part of a desire to collaborate with other developers and to move the IT community forward collectively. She also started the blog zootopi.dev to share her technological knowledge and tips with as many people as possible. In terms of scientific events, she was, for example, co-chair of the ESSLI 2024 Summer School Student Session, program committee member, reviewer of seven international workshops and conferences, and organizer of a conference in France.

Volunteer teaching: Being convinced that education is essential to reduce inequalities and open doors to scientific careers for people from diverse backgrounds, she dedicated part of her time to volunteer teaching at non-profit organizations. She taught 5 courses for over 1,200 students at VietAI, a non-profit organization that aims to develop AI talent and create a community of world-class AI experts in Vietnam. She was one of the organizers of an online seminar titled "Women and Artificial Intelligence: What opportunities for women?", which brought together around 120 participants and aimed to raise public awareness of the place of women in this rapidly developing field.