Towards Extracting and Utilising Entities in Task Specific Low Resource Settings

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communication Engineering by Research

by

Swayatta Daw
2020702016
swayatta.daw@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Towards Extracting and Utilising Entities in Task Specific Low Resource Settings**" by **Swayatta Daw**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Advisor: Prof. Vikram Pudi

Dedicated to

my grandmother, Late Mrs. Namita Dasgupta,

for setting an example of excellence, both in academics and in life,

and to my parents, Mrs Saswati Daw and Mr Aloke Daw,

for being by my side always

Abstract

The task of entity extraction has been thoroughly explored in the NLP community over the last two decades. There is a myriad of downstream tasks for Natural Language Understanding that utilizes entity extraction - ranging from Information Retrieval, Question Answering, Fact Extraction and Verification, Knowledge Graph Completion, etc. However, the datasets and domains that have been used for evaluation and benchmarking such entity extraction models have mostly been straightforward, structurally simple, semantically non-complex, and well-present across the breadth of training data. Most entity extraction tasks comprise of a significant overlap in entities across train and test sets. This does not mimic the real-world scenario, where rare emergent entities show a larger presence, and the entities themselves are often complex and semantically ambiguous. Our work investigates entity extraction in 2 settings - in the domain of scientific research papers, and in the area of linguistically low-resource structurally complex settings. In the first scenario, we establish an end-to-end pipeline that extracts certain task-specific entities from research documents, that help in a meaningful mapping of the research landscape. We show that the domain of scientific documents is non-trivial for entity extraction tasks because scientific entities follow a long-tail distribution. We incorporate multiple strategies like Distant Supervision, Graph Ranking, and Sequence Labelling to solve this task. Furthermore, we introduce multiple human-annotated gold standard datasets for both modular and complete evaluation of this entire pipeline. In the second setting, we investigate the task of low-resource semantically ambiguous complex entities. We experiment with multiple transformer-based architectures and pre-training strategies to solve this task. Our work significantly outperforms the baseline and outperforms multiple ensemble and gazetteer-based systems. We hope that our work will help undertake further research in this critical area of Natural Language Processing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The task of entity extraction in Natural Language Processing has attracted significant research interest since its inception. The task entails the extraction of factually relevant entities from unstructured data and their classification into relevant predefined categories and subcategories. Such extracted entities often have a myriad of applications in various downstream tasks concerning Natural Language Understanding. Their usage spans across various domains - including Material Science [65], Biomedical data [31], Legal documents [15], Scientific Publications [59] etc.

Our work focuses on Extracting and Utilising entities across two settings - scientific documents, and semantically ambiguous settings for low-resource languages. In the subsequent sections, we elaborate on the key motivations for our research direction, the formal definitions of the tasks undertaken, and the contributions of our research work in these tasks.

1.1.1 Exponential Increase in the Quantity of Scientific Publications

The number of publications of research papers have increased exponentially in the recent past. This is especially true in the domain of Computer Science, which has witnessed a significant rise in research interest over the last two decades. Hence, it has become increasingly laborious for researchers to stay abreast of the recent developments across the breadth of their research interest. Our motivation comes from the need of creating a map of the entire research landscape of Computer Science, such that the state-of-the-art developments are captured in an automated manner.

Papers with Code (PwC[1]) is a community driven platform that updates the recent developments in various tasks of different areas of Computer Science. They segment each papers into various categories, like Tasks, Datasets, Method and captures where the target paper falls

[1]https://github.com/paperswithcode/paperswithcode-data

as compared to other similar works for the same task. They benchmark it manually, using the results obtained from the paper. Our motivation is to create an automated counterpart of Papers with Code, where the models used in a paper get automatically benchmarked in a scalable and efficient manner. The work presented in this thesis serves as starting point for this project. More specifically, our work aims to extract competing model entities from a research document. The formal details of this task are covered in Section 1.2.

1.1.2 Low Resource and Semantically Ambiguous Scenarios for Named Entities

The research community has predominantly dealt with named entities that are relatively easier to extract because of multiple reasons - they have a significant overlap between train and test sets [3], they are obtained from well-defined categories, and the entity structure contains words that are named entities themselves.

However, under practical scenarios, high-performing models in the former setting fails [45]. Entities are often structurally complex [1], semantically ambiguous and occur in a zero or few-shot setting, with the model having to deal with largely unseen entities during inference. The detection of such entities becomes non-trivial. Our work aims at building NER models for such scenarios, without using any external gazetteer or ensembling technique.

1.2 Extraction of Competing Models using Distant Supervision and Graph Ranking

We introduce the task of extracting competing models from research documents. We treat model names as named entities occurring in a scientific document. We establish an end-to-end pipeline that extracts all the competing model names from a research document. Competing model names are defined as those model names that solve the same task as outlined in the research paper. They contribute directly in solving the same task as outlined by the research objective present in the research paper. This objective may have been shared by multiple other research papers, which the current target document cites. The model names are usually mentioned in the cited text span of the document. We extract these model names as competing models. There are other multiple model named entities present in the document that contribute indirectly in solving the task outlined. Hence, they are termed as non-competing. The task is non-trivial because we require the document context as a whole to extract these entities. On top of that, model names themselves follow a long-tailed distribution. We show empirically that this leads to overfitting or memorisation of model names by the sequence labellers. Hence, we need effective strategies to deal with the long-tailed entities. We use multiple strategies as part of our pipeline to circumvent the aformentioned challenges. We crawl a publicly available knowledge

base to gather an exhaustive list of model named entities. We use a large unlabelled corpus of research papers, along with the list of model named entities, to create a distantly supervised training set. We rely only on this training set to train the sequence labelling model that detects model names from research papers. Furthermore, we annotate 1000 sentences to evaluate our trained sequence labeller. Now, in order to capture the whole document context, we utilise unsupervised graph-ranking alogorithms for key-phrase extraction. We combine the graph-ranker and the sequence labeller to capture both the document context and the local sentence context to predict model names that are competing for the same task. We also annotate an exhaustive list of papers to create the gold labels for our evaluation of the entire pipeline.

1.3 Transformer-based Architectures for Complex NER

Complex entities can be broadly classified into 3 categories - linguistically complex (*"Eternal Sunshine of the Spotless Mind"*), semantically ambiguous (*"Among Us"*) and emerging, largely unseen, newly added entities. We use the MultiCoNER dataset [41] for our task. This dataset is challenging in nature because of the large amount of complex entities present in the dataset. There are a variety of different types of named entities present which are either structurally complex, semantically ambiguous and emerging in nature. Hence, a large number of pre-trained language models don't have prior exposure to such named entities. Apart from this, they ensure that there is not much overlap in the train and test set of the entities. This is particularly prevalent because of the sheer mismatch in scale of train and test data. The quantity of test data outnumbers that of train by over 100 times. This difference in scale is helpful in mimicking the real world settings, where in production scenarios, models may encounter a variety of different types of data due to the sheer scale of the real world. Hence, this dataset is one of a kind dataset where the number of training samples are significantly less than the evaluation samples. This ensures a rigorous evaluation and provides a greater emphasis on generalisation. Simply using a large amount of gazetteer collected training data defeats the purpose of such a setting, as the whole point is to build models that can robustly generalise across the production setting while being trained on datasets which have significantly lesser amount of samples. In our work, we show that the usage of simple BERT-based sequence labellers work surprisingly well for such tasks. The attention mechanism present in BERT along with a classification head on top serves as a robust model which generalises well across the larger test set. Our method successfully beats a number of other pariticipants who utilise external gazetteers or ensemble methods for their approaches. Our method also outperforms the baseline by significant margin.

We experiment with multiple architectures for BERT-based models. We try BERT-Linear, BERT-CRF and BERT-BiLSTM-CRF based models for our experiments. For the English dataset, we find that the BERT-Linear model works the best as compared to other architectures. This is because BERT is already pre-trained on a vast amount of English corpus. A simple linear

layer prevents overfitting on the smaller training set. We observe the heavier layer performs significantly because they add no value on top of the already pre-trained BERT embeddings, and their lack of generalisation over the larger test set.

For the lower resource languages like Spanish and Chinese, we use the strategy of Whole Word Masking (WWM). Vanilla BERT uses sub-word masking where it masks a subtoken of a work for Masked Language Modelling. However, this strategy is ineffective for morphologically rich languages like Chinese, because a Chinese character by itself imparts a specific meaning. Also, it is significantly easier for a Language Model to decipher the mask for sub-word masking. With Whole Word Masking strategy, it is more difficult for the LM to predict the masked token. Hence, the training is more robust in nature. We find the that Whole Word Masking strategy improves performance across the board, where the BERT-CRF model performs the best across all architectures. We find that BERT-CRF performs best because unlike in English, the Spanish and Chinese pre-trained BERT does not contain enough training information and needs heavier layers to drive the train loss down.

1.4 Key Contributions and Thesis Outline

1. We perform an elaborate survey of the research landscape of named entity recognition in scientific documents, NER in low resource, complex and structurally ambiguous settings, pre-trained language models that aid in the task of NER and unsupervised graph labellers that fulfill a crucial part of the competing model extractor pipeline. We elaborate on the related work in Chapter 2.

2. We introduce the task of extraction of competing models. We use multiple techniques like distant supervision to tackle the label scarcity problem. We use graph ranking to capture the entire document context. We introduce a simple entity replacement technique to counter overfitting and entity memorisation for long-tailed distributed entities. We also introduce two distinct evaluation sets for evaluating the sequence labeller and the entire pipeline. We finally achieve a good F-1 score that enables our model to be treated as a baseline while leaving room for further research work. We provide entire details of this work in Chapter 3.

3. We explore the task of Complex NER in English language. Complex NER comprises of detection of named entities that are structurally complex in nature, semantically ambiguous and rarely occur in existing training datasets. We use the MultiCoNER dataset which is specifically designed for such types Complex Named Entity Recognition task. We experiment with multiple transformer-based architectures like BERT-Linear, BERT-CRF and BERT-BiLSTM-CRF. We show that simpler models generalise better and are able to beat larger ensemble models. We are also able to beat systems trained on extra

gazetteer-based training data. Our models significantly outperform the baseline and are able to achieve a competitive ranking in the shared task of MultiCoNER, comprising of multiple international teams. We provide details of this work in Chapter 4.

4. We investigate the task of detection of Complex Named Entities under a low resource setting. The definition of complex named entities stays the same, that is, they are either structurally complex, semantically ambiguous or emerging in nature. We investigate such named entity detection under a low resource language setting, using Spanish and Chinese as our chosen language. We investigate multiple BERT-based architectures for our experiments. We also show that the strategy of Whole Word Masking (WWM) works better for morphologically rich languages like Chinese. Our model is successfully able to beat systems utilising external gazetteers and ensemble-based models. We provide the details of this work in Chapter 5.

5. Finally, we conclude our findings in Chapter 6, and discuss about the possible threads of research direction in these areas.

Chapter 2

Related Work

In this chapter we highlight the prominent research directions undertaken by the community. We discuss multiple areas of research that have been utilised in different facets of our work; either as a method or as a task itself. Since our work primarily revolves around Named Entity Recognition, we put special emphasis on this area of research. We focus primarily on the Sequence labelling approaches for Named Entity Recognition, the pre-trained language models primarily used for NER, occurence of Named Entities in Scientific Documents, and the emerging research direction of low-resource NER. We then move on to another related area of research, which is the utilisation of unsupervised graph-ranking algorithms for key-phrase extraction. We use key-phrase extraction as a crucial module in our pipeline for the task of extraction of competing models from research documents.

2.1 Sequence Labelling for Named Entity Recognition

Sequence Labelling is a task in Natural Language Processing that seeks to classify each token in a sequence, by assigning a label to each such token. The token can be considered the smallest meaningful subpart that makes up the larger sequence. The way a sequence is divided into tokens is decided by the task definition. Each token is assigned a particular label after classification, where the label space is derived in way such that the meaning of the assigned labels depend on the specific task type. The task types are generally divided into types suck as : Part-of-Speech (POS) tagging [5], Named Entity Recogntion (NER) [48], text chunking [54], co-reference resolution [32] and relation extraction [50]. Named Entity Recogntion (NER) is a crucial task of Information Extraction where we seek to extract Named Entities (NE) from a given context. In general named entity tasks, we generally classify the extract Named Entities into different labels - like Person, Organisation, Geographical Locations etc. However, this label space is heavily determined by the objective and domain of the task in question.

NER has traditionally been treated as a sequence labelling problem, using CRF [28] and HMM [11]. Recent approaches have used deep learning based models [33] to address this task,

which require a large amount of labelled data to train. The high cost of labelling remains the main challenge to train such models on rare long tailed entity types, where availability of labelled data is scarce. In order to address the label scarcity problem, several methods like Active Learning [20], Distant Supervision [61, 35, 23], Reinforcement Learning-based Distant Supervision[49, 70] have been proposed. [37] focused on detecting dataset mentions from scientific text and used data augmentation to overcome the label scarcity problem.

A widely used benchmark for NER was the CoNLL 2003 shared task. It contained annotated newswire text from the Reuters RCV1 corpus. Previous researchers [4] had used BiLSTM models with attention to predict named entities on this dataset. [40] use a BiLSTM-CNN-CRF to predict the named entities.

2.2 Pre-trained Language Models for NER

Ever since the introduction of BERT[14], transformer based pre-trained language models have effectively utilised transfer learning for downstream NLP tasks. NER has been traditionally modeled as a sequence labelling problem. [24] proposed a Bidirectional LSTM with a CRF layer on top for classifying tokens as entities. [25] use a pretrained BERT model with a CRF layer on top for performing NER on DailyHunt news dataset. We use a BERT-based model with a CRF layer on top and achieve competitive performance on low-resource NER tasks on multiple languages, beating the baseline by a significant margin in each case.

2.3 Named Entities in Scientific Documents

Long tailed entities are named entities which rarely occur in text documents. For these types of entities, the task of Named Entity Recognition (NER) is non-trivial. Recent approaches have aimed at solving the problem of NER using supervised training using deep learning models. However, supervised learning techniques require a large amount of token-level labelled data for NER tasks. Annotating a large number of tokens can be time-consuming, expensive and laborious. For real-life applications, the lack of labelled data has become a bottleneck on adopting deep learning models to NER tasks.

Most scientific named entities can be classified as long-tailed entities because of the rarity and domain-specificity of their occurrence. Recent work on NER in scientific documents has been concentrated around detecting biomedical named entities [27] or scientific entities like tasks, methods and datasets [38, 26, 46]. Some papers like [37] focus on the detection of a single specific entity-type (like dataset names) from scientific documents. Although previous work has focused on identifying methods [38, 26] as named entities, but what constitutes a method can have a significant variance when it comes to human annotated data. The authors

[38] report the Kappa score of 76.9% for inter-annotator agreement in the SciERC dataset, which is widely used as a benchmark for scientific entity extraction.

2.4 Low Resource NER

The task of low-resource NER has been investigated before by multiple researchers. This line of research focuses mainly on leveraging the cross-lingual contextual information obtained from low resource languages. [16] use cross-lingual knowledge transfer to train the NER model for the low- resource target language. [67] use bilingual dictionaries to tackle the task of low-resource NER. [53] proposes a Bayesian graphical model approach to improve performance on NER tasks. NER models often use gazetteers (list of named entities) to improve performance in NER tasks. [56] creates soft-gazetteers for low-resource languages, leveraging English Knowledge Bases. [6] focuses on an unsupervised approach for NER for to circumvent the label scarcity problem in low-resource languages. [52] leverages multilingual transfer learning from multiple languages for low-resource NER tasks. [23] uses distant supervision in the low-resource setting for NER.

There are multiple approaches that have been undertaken in the recent past to improve the state-of-the-art in NER tasks. [62] uses concatenation of embeddings to outperform the state-of-the-art in NER tasks, as they infer that concatenation of embeddings lead to a better word representation. Their method automates the process of finding meaningful embeddings to concatenate for improved performance. [74] propose a co-regularization framework for entity extraction comprising of multiple models with different architectures but different parameter initialisations. This helps to tackle overfitting of large neural network-based model on low-resource training data for NER. [58] use document-level features to improve information extraction on entity-centric tasks. NER and Relatation Extraction are the core information extraction tasks in NLP. [72] models this as a span-pair classification problem, and they further improve the pair representations by considering the dependencies between the spans (pairs) by strategically packing the markers in the encoder. [69] proposes a novel entity-aware self attention framework for transformer based models for NER. [64] extracts document-level context for sentences for which document information is absent. They treat the sentence as a query and uses a search engine to extract the document level contextual information. [39] uses multiple neighbouring sentences as the contextual information for NER.

2.5 Unsupervised Ranking Algorithms for Keyphrase Extraction

Key-phrase extraction [22] is a task in NLP that seeks to extract phrases that convey factually relevant information from a certain context. The goal of key-phrase extraction is to extract

a set of phrases that are related to the main topics discussed in a document. In this section, we discuss the various unsupervised approaches for key-phrase extraction. EmbedRank[8] is an unsupervised algorithm that utilises the cosine similarity between semantically similar representations to extract key-phrases. It extracts candidate phrases based on POS sequences and uses sentence embeddings (Doc2Vec or Sent2vec) to represent both the candidate phrases and the document in the same high-dimensional vector space and ranks them using cosine similarity with respect to the document embedding. [73] propose WikiRank, an unsupervised automatic keyphrase extraction method that links semantic meaning to text. In graph-based ranking algorithms, candidate phrases are treated as nodes and related candidate phrases are connected by edges. TextRank [47] considered related candidates as co-occurring phrases within a given window. It is a variant of the PageRank algorithm that considers each node as the candidate key-phrase. An edge occurs between two nodes if they appear within a given window. The edges are unweighted. These nodes and edges are used to form a graph, upon which the PageRank algorithm is run to rank the nodes.

SingleRank [60] is a modification of TextRank that added weights to the edges between related candidates. The weights are inversely proportional to the distances between the occuring nodes in the document. This incorporated a form of positional information that improves TextRank to some extent. SGRank [13] and PositionRank [17] incorporated statistical and positional heuristics into a graph-based algorithm to obtain ranked keyphrases. MultipartiteRank [9] is an advanced version of TextRank that incorporates positional knowledge in edge weights, leading to state-of-the-art performances over benchmark datasets. It uses agglomerative hierarchical clustering to cluster key-phrases based on topics. This clustering is performened based on the stem form of the key-phrase. They create a multipartite graph that ensures topical diversity in the ranking mechanism. The first occurring key-phrases from a topic in the document are imparted higher weightage. This leads to the preservation of topic diversity and positional information, leading to state-of-the-art results over multiple forms of datasets, performing especially well for scientific documents. The reason being scientific documents are longer in nature, and the topical clustering mechanism along with the positional information captures the factually relevant keyphrases while eliminating redundancy.

Chapter 3

Extraction of Competing Models from Scientific Documents using Distant Supervision and Graph Ranking

3.1 Overview

We introduce the task of detection of competing model entities from scientific documents. We define competing models as those models that solve a particular task that is investigated in the target research document. The task is challenging due to the fact that contextual information is required from the entire target document to predict the model entities. Hence, traditional sequence labelling approaches fail in such settings. Furthermore, model entities themselves are long-tailed in nature, i.e, their prevalence in scientific literature is limited, along with a scarcity of labelled data for training supervised learning techniques. To address the above bottlenecks, we combine an Unsupervised Graph Ranking algorithm with a SciBERT-CRF based sequence labeller to predict the entities. We introduce a strong baseline using the above mentioned pipeline. Also, to address the label scarcity of long-tailed model entities, we use distant supervision leveraging an external Knowledge Base (KB) to generate synthetic training data. We address the problem of overfitting in small sized datasets for supervised NER baselines using a simple entity replacement technique. We introduce this model as part of a starting point for an end-to-end automated framework to extract relevant model names and link them with their respective cited papers from research documents. We believe this task will serve as an important starting point to map the research landscape of computer science in a scalable manner, needing minimal human intervention.

3.2 Introduction

The number of scientific publications in the computer science domain has increased exponentially in the recent past. Hence, it has become increasingly cumbersome for researchers to keep track of the advancement of the research landscape. Often, research papers introduce new

models that perform strongly in comparison with the baseline or advance the state-of-the-art. In order to effectively benchmark models and compare their performances, it is important to be able to map the research landscape for similar or related tasks. Papers with Code (Pwc[1]) is a community driven corpus that serves to automatically list models that solve particular subtasks , with links to the scientific research paper that introduced the model. Our aim is to build a similar but automated end-to-end pipeline which detects model names from scientific papers and benchmarks them against other similar models that solve the same task.

Long tailed entities are named entities which rarely occur in text documents. For these types of entities, the task of Named Entity Recognition (NER) is non-trivial. Recent approaches have aimed at solving the problem of NER using supervised training using deep learning models. However, supervised learning techniques require a large amount of token-level labelled data for NER tasks. Annotating a large number of tokens can be time-consuming, expensive and laborious. For real-life applications, the lack of labelled data has become a bottleneck on adopting deep learning models to NER tasks.

Most scientific named entities can be classified as long-tailed entities because of the rarity and domain-specificity of their occurrence. Recent work on NER in scientific documents has been concentrated around detecting biomedical named entities [27] or scientific entities like tasks, methods and datasets [38, 26, 46]. Some papers like [37] focus on the detection of a single specific entity-type (like dataset names) from scientific documents. Although previous work has focused on identifying methods [38, 26] as named entities, but what constitutes a method can have a significant variance when it comes to human annotated data. The authors [38] report the Kappa score of 76.9% for inter-annotator agreement in the SciERC dataset, which is widely used as a benchmark for scientific entity extraction.

In this work, we introduce the task of extracting competing model names from a research paper. Model name entities themselves follow a long-tailed distribution. So, we establish an end-to-end pipeline that extracts all the competing model names from a research paper and links them to their respective citation. We also experiment with other pipelines for qualitative analysis of our work.

While browsing related work for a given task, a researcher has to manually visit every research paper that uses a competing model that is used for the same task. This process is time-consuming if a survey of a research landscape is to be done on a large scale. Our motivation is to automate this process by automatically extracting model names that solve a similar task and linking them to their corresponding cited paper. If executed on a large scale, this pipeline would be able to effectively map the computer science research landscape in an automatic and scalable manner with minimal human intervention.

We introduce a strong baseline for this task by combining an unsupervised document level graph ranking algorithm and a supervised BERT-based sequence tagger to obtain entity model

[1]https://paperswithcode.com/

names. Essentially, we treat the relevant keyphrases extracted by the graph ranker as a superset of candidates for the sequence labeller.

We introduce two datasets for this task. For training the supervised sequence tagger, we create weakly supervised distant labels using an external Knowledge Base and unlabelled corpora. We also release a manually annotated dataset for the evaluation purpose of the sequence tagger. For evaluating the entire framework of competing model name extraction, we release another dataset with full paper document level annotation. Furthermore, we use a simple entity citation linking technique to link the extracted model names with their respective citation in the research document. We believe this task will be a significant step forward towards mapping the research landscape of computer science.

Our contributions can be summarised as follows:

- We introduce a novel approach of treating ranked keyphrases as a superset of sequence labellers for solving this task. To the best of our knowledge, this approach has not been used before in prior research work. We believe this approach can be extended to other similar tasks that require document level contextual information for NER.

- We create an annotated dataset of annotated full papers for evaluation of the pipeline. Previous datasets for sequence labelling in the scientific literature focused only on annotating abstracts of scientific papers [38, 26]. We believe the approach of incorporating full length document information is crucial to capture the entire document context, hence we introduce a full paper annotated dataset for final evaluation.

- We introduce strong baselines while relying only on distantly supervised weak labels to train our sequence labeller. We evaluate the trained model on our annotated evaluation dataset.

3.3 Motivation

Papers with Code (PwC[2]) is a community driven corpus that serves to automatically list models that solve particular subtasks, with links to the scientific research paper that introduced the model. Our aim is to build a similar but automated end-to-end pipeline that detects model names from scientific papers and benchmarks them against other similar models that solve the same task. We believe the task introduced in this paper (extraction of competing model names from scientific documents) to be a significant step forward towards the whole pipeline.

[2]https://github.com/paperswithcode/paperswithcode-data

In this paper, we present SDP-LSTM, a novel neural network to classify the relation of two entities in a sentence.

Inspired by the unique feature representation learning capability of deep autoencoder, we propose a novel model, named Deep Autoencoder-like NMF (DANMF), for community detection.

We introduce the Multi-View Transformation Network (MVTN) that regresses optimal view-points for 3D shape recognition, building upon advances in differentiable rendering.

Figure 3.1: Example sentences with annotated model name entities

3.4 Task Definition

We define **competing** models as model names that attempt to solve the same task as investigated by the target research paper. For example, if a research paper investigates the task of producing knowledge base embeddings, TransR [36] will be a competing model name as it has been introduced by prior research work to solve the same task. If a research paper investigates the task of Question Answering, some competing model names can be T5 model [51] or XL-Net [71], because these are models that have been used to solve this task in prior research work. A non-competing model name would be a model that has not been used directly to solve the same task. We provide a few examples to illustrate the difference between a competing and a non-competing model in Table 3.1. For the first two examples, the models highlighted in bold are competing models because they directly solve the task investigated in the input research paper. For the third example, TransE is a competing model, but Word2Vec is not. The reason for this is that TransE produces Knowledge Base embeddings directly that aid in Knowledge Base completion (which is the target task in the research paper). But, Word2Vec is a language model that TransE is inspired by, as denoted in the sentence. Hence, it only contributes indirectly to the research task. So, it is a non-competing model. Similarly, HyperOpt, in the last example, is non-competing, as it is an algorithm the authors used for hyperparameter search and is not a model that contributes directly in solving the task investigated in the input research paper.

Our task in this paper is to detect competing model names given an input research document. Also, after extracting the model names, we link the extracted entities with their respective cited papers.

Type	Sentence	Paper Title
Competing	Other transition-based models extend **TransE** to additionally use projection vectors or matrices to translate head and tail embeddings into the relation vector space, such as: **TransH** (Wang et al., 2014), **TransR** (Lin et al., 2015b), **TransD** (Ji et al., 2015), **STransE** (Nguyen et al., 2016b) and **TranSparse** (Ji et al., 2016).	A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network
Competing	In Table 2, we compare **SCIBERT** results with reported **BIOBERT** results on the subset of datasets included in (Lee et al., 2019).	SCIBERT: A Pretrained Language Model for Scientific Text
Non-competing	**TransE** [4] is a translation based model inspired by *Word2Vec* [16]	On Evaluating Embedding Models for Knowledge Base Completion
Non-competing	(Xie et al.2016) use a variety of models including *convolutional neural networks (CNN)* to encode word sequences in entity descriptions.	KG-BERT: BERT for Knowledge Graph Completion
Non-competing	To find the hyper-parameters, we used *HyperOpt* (Bergstra et al., 2015), which uses Bayesian optimization.	Tabular Data: Deep Learning is Not All You Need

Table 3.1: **Few examples of competing and non-competing models. The competing models are highlighted in bold, whereas the non-competing models are highlighted in underlined italic.**

3.5 Annotation Process

We create two datasets for training and evaluation. We annotate sentences from scientific papers as per token-level BIO tagging scheme to evaluate our sequence labeller, which only

	Train	**Test**	**Total**
# sentences	7800	1000	8800
# tokens	232600	22873	255473
# entities	19012	3647	22659
# unique entities	14748	1249	15672
avg # tokens per sentence	29.82	22.873	29.03
avg # entities per sentence	2.44	3.65	2.57

Table 3.2: Overall statistics of train and evaluation dataset

uses contextual information from a input sentence for sequence tagging. To evaluate the whole pipeline, we provide document-level annotations with full length research papers as input and competing model names as the annotated output. We use two different datasets for a more comprehensive evaluation, as our pipeline uses two stages. The first stage involves extracting candidate keyphrases utilising the entire document level information for keyphrase ranking. The second stage is our sequence labeller that uses sentence level information to find model named entities. We describe the annotation process for the dataset creation for sequence labelling first. Considering our end goal of automating a high precision framework of extracting related model names and to minimise ambiguity ,we consider only named models as model entities for this task . Few examples are - *NMN+LSTM+FT, SpERT (with overlap), B-BOT + Attention and CL loss, SA-FastRCNN, DS-CNNs (Random Walk), Sparse Transformer 59M (strided).* We consider model entities that have an unique name or that are formed by combination of other model names, eg - NMN+LSTM+FT. A few example sentences with model entities are displayed in Figure 3.1. We define and annotate the test corpus using the standard BIO tagging scheme. Each model entity type was defined to have maximum span length. For Acronyms, we consider the full length entity name instead of the short form acronym if it occurs in text - eg. *DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations.* On average there are 2.5 tokens per entity. We refer Google Scholar and Semantic Scholar to confirm entity types. We randomly selected a subset of abstracts from the arxiv dataset containing 1.7M+ paper data and metadata and randomly select sentences from them to annotate. Also, we randomly sample the DBLP citation dataset containing 1,511,035 papers and obtain the full length versions from the available papers using DOI matching and obtained a random sample of sentences from the full text. We use two different sets of corpus because we want our model to be evaluated on multiple domains within computer science and different publication venues. All the statistics related to our annotated corpus and train set are provided in Table 3.2

For evaluating the whole pipeline, we annotated full length research papers. We read through the introduction and find out the task the paper solves. Then we browse the entire paper and

find all mentions of model names that solve a similar task. The process has a low level of ambiguity because a majority of the model mentions occur in the related work section,citation contexts or experimental results section. It is a standard practice among authors to cite the relevant research paper if they mention any model names from prior research work. Hence, we only consider models that the authors cite to be candidates for competing models. We make sure the labelled entities are model names by referring to Google Scholar and Semantic Scholar. If there is any ambiguity regarding whether a labelled entity is a model name or not, we discard the full paper. To infer if a model is a competing model or not, we find the task or the problem the paper solves. This is usually mentioned clearly in the introduction and the related work section. We label the model entities (that the authors mention as solving a similar problem or task as the original paper) as competing models. To further verify that the claim by authors is indeed true, we visit the cited research paper and ensure that the model is solving a similar task. Furthermore, we only consider papers where the "competing" relation among models is clear and discard any paper where there is ambiguity regarding this relation. Hence, we ensure ambiguity to be significantly low regarding our annotations. The statistical details about the annotations are provided in the Table. As we ensure a negligible level of ambiguity, we use only one human annotator (one of the authors in this paper) for our annotation process. We believe the need of multiple annotators for an inter-annotator agreement is insignificant for our task, as a low level of ambiguity is ensured by considering only named models and clearly defined tasks with competing model names.

3.6 Method

Our entire pipeline has two components. Firstly, we extract all citation sentences from the input research paper. We combine all the citation sentences to create a mini-document. We use a graph ranking algorithm to extract all the candidate keyphrases from this mini-document. This graph ranking algorithm utilises document level information to rank keyphrases. Secondly, we use a sequence labeller for extracting named entities from the positively labelled citation sentences. Lastly, we merge the results of the graph ranker and the sequence labeller to output final competing model entities. In section 3.6.2, we provide details about the training process and the model for our sequence tagger. In section 3.6.1, we provide details about the unsupervised graph ranking algorithm for keyphrase extraction.

3.6.1 Graph-Ranking Algorithm

We use Multipartite Rank [9] as it had proved to be the state-of-the-art among all keyphrase ranking algorithms, performing particularly well on longer scholarly documents. We briefly describe how we use this algorithm for unsupervised keyphrase extraction.

Let C be the set of all citation sentences in a document d. C forms an order set of citation sentences, which is collectively treated as a document. We build a graph representation of C. A set of candidate keyphrases K is extracted from C. The candidate keyphrases K are grouped into topics based on the stem forms of the words they share using hierarchical agglomerative clustering with average linkage. The candidate keyphrases are used to build a multipartite graph, where the nodes are keyphrase candidates that are only connected if they belong to a different topic. The edges between each node is weighted as the inverse of the distance between the two keyphrases K_i, K_j in C. Weight wij is calculated as the sum of the inverse distances between Ki and Kj:

$$w_{ij} = \sum_{p_i \in P(K_i)} \sum_{p_j \in P(K_j)} \frac{1}{p_i - p_j}$$

where $P(K_i)$ is a set of word offset positions of K_i. The first occurring candidates of each topic are promoted more as they capture higher relevance. Weights of the first occurring candidates of each topic is modified according:

$$w_{ij} = w_{ij} + \alpha . e^{\frac{1}{p_i}} \sum_{K_k \in T(K_j) \setminus K_j} w_{ki}$$

where α is a hyperparameter that controls the strength of the weight adjustment, $T(K_j)$ is the set of candidates belonging to the same topic as K_j , p_i is the offset position of the first occurrence of candidate K_i. After the graph is built, a ranking algorithm is then used to order each keyphrase candidate K_i. We adopt the popular TextRank Algorithm [47] for the ranking mechanism. A final set of top ranked keyphrases $\tilde{K}$ is obtained.

3.6.2 Sequence Tagging

For training our sequence tagger, we only rely on distant labels created using an external Knowledge Base and an unlabelled research text corpus. We also demonstrate that for long-tailed entity type, there is a need to ensure fairer distribution among entity occurrence, in order to prevent overfitting, which occurrs in the form of the model memorising certain popular entity names. The details about the training set creation is provided in section 3.6.2.1. The details about the model and the results on the evaluation set is provided in section 3.6.2.2.

3.6.2.1 Training Set Creation with Entity Replacement

We utilise the publicly available Papers with Code (PwC) corpus as a Knowledge Base. We crawl PwC and obtain all the model names occurring in the metadata for each task and subtask. We obtain a total of 14,748 model names. For the unlabelled corpora, we use a total of $\tilde{2}$27,000 abstracts from arxiv and obtain all sentences (7800) containing a model name mention. We find that the occurrence of some model names are much more frequent in literature (e.g - CNN).

Due to the small dataset size and the large imbalance in few entity mentions, the model is prone to overfitting. To mitigate this, we use a simple entity replacement technique, where we find all model entity mentions, and randomly replace them with other names to ensure a fairer distribution. The distribution pre-replacement is shown in Figure 3.2. We use all 14,748 model entities at least once and limit an entity occurrence to at most 2 in the train dataset, after replacement.

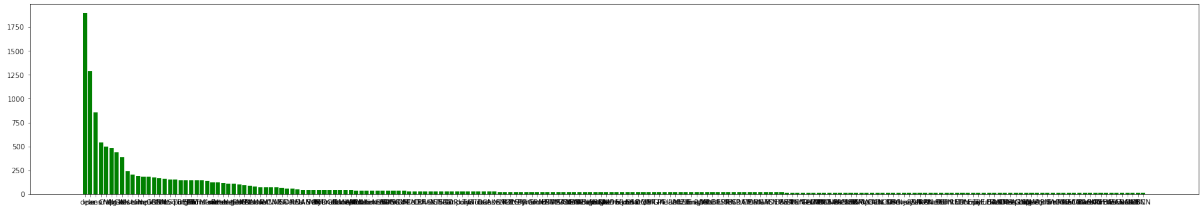

Figure 3.2: Distribution of entity occurrence frequency in the training dataset pre-replacement

3.6.2.2 Distantly Supervised NER Model

We treat NER as a sequence labelling problem. Given a sequence of N tokens $X = [x_1, ..., x_N]$, we aim to find an entity which is a span of tokens $s = [x_i, ..., x_j](0 \leq i \leq j \leq N)$ associated with the entity type model name. We formulate this as a sequence labelling task of assigning a sequence of labels $Y = [y_1, ..., y_N]$. The aim of our sequence labeller is to classify each token as a certain entity type as per the BIO tagging scheme.

We consider K train sentences denoted as $\{(X_k, Y_k)\}_{k=1}^{K}$ with distant token level annotations. We aim to learn a function $f(X, \theta)$, which can correctly predict the entity labels for a train sentence X_k. We minimise the loss:

$$\theta^* = \arg\min_{\theta} \frac{1}{K} \sum_{k=1}^{K} l(Y_k, f(X_k, \theta))$$

over $\{(X_k, Y_k)\}_{k=1}^{K}$ where θ is the parameter and l is the cross-entropy loss.

We experiment with multiple baselines which are standard for the sequence labelling process.

- A **BiLSTM + CRF** model where the bidirectional contextual representations are captured by the BiLSTM model, and the resultant representations are passed to the Conditional Random Field (CRF) that produces sequence labels as output.
- A **BERT + CRF** model where the contextualised embeddings are captured by a pre-trained BERT base uncased model and passed onto the CRF layer to produce token labels.
- A **SciBERT + CRF** model where the domain specific contextualised embeddings are captured by a pre-trained SciBERT [7] model. SciBERT is BERT-based language model

train on large unlabelled scientific corpora using MLM objective. The output embeddings are passed to the linear CRF layer which predicts token labels from contextual representations.

We evaluate our baselines using our evaluation dataset and the results are displayed in Table 3.3. We demonstrate that entity replacement provides a significant boost in performance for each of these models. The reason is that the model does not memorise entity names for the replaced dataset and uses the context to predict the entity types. The results also prove that standard NER approaches can provide decent results on the evaluation dataset while relying only on weakly labelled training data.

3.7 Combining Graph-Ranker and Sequence Tagger

We used the Unsupervised Keyphrase Extraction algorithm to capture only those keyphrases that are most relevant to the document. Although the Sequence Tagger performs well on detecting model name mentions using sentences as the contextual information, we need to capture document level relevance as well to extract competing models. The reason is that not all model name mentions are relevant to the task the given target research paper aims to solve. Hence, we predict only those entities which are common to both top-ranked keyphrases and the extracted model names from our distantly supervised sequence tagger. More formally,

$$Y'' = \tilde{Y} \bigcap \tilde{K}$$

where $\tilde{Y}$ is the set of predicted entities by the sequence tagger, $\tilde{K}$ is the set of top-ranked keyphrases and Y'' is the final set of predicted entities.

3.8 Results

We experiment with different Sequence Labellers and show the results in Table 3.3. We also show that our entity replacement technique contributes to a major boost in overall performance of the labellers. These Sequence labellers are evaluated on human annotated gold labels, with sentence level annotations. For the evaluation of the entire combined pipeline, we use the full-length document-level annotated dataset. We find that MultiPartite Rank combined with SciBERT-CRF (with replacement) gives the best performance on our task.

3.9 Sentence Intent Classification

Apart from the intersection of Graph-Ranker and Sequence Labeller, we also experiment with Sentence Intent Classification. We train a classifier to detect whether a sentence contains

	P	R	F1
BiLSTM + CRF (w/o replacement)	0.205	0.519	0.294
BERT + CRF (w/o replacement)	0.389	0.310	0.345
SciBERT+CRF (w/o replacement)	0.391	0.312	0.346
BERT+CRF (with replacement)	0.575	0.563	0.569
BiLSTM + CRF (with replacement)	0.628	0.631	0.629
SciBERT+CRF (with replacement)	**0.641**	**0.632**	**0.636**

Table 3.3: Result on Evaluation Dataset

	P	R	F1
SciBERT-CRF	0.290	0.764	0.420
Multipartite Rank	0.123	**0.834**	0.214
Multipartite Rank+SciBERT-CRF	**0.639**	0.672	**0.655**

Table 3.4: Final Evaluation of the Combined Graph-Ranker and Sequence Tagger

relevant information regarding models that solve a similar task as specified in the target research paper. For a target scientific document, we define a relevant model name as a model that the author has cited, which solves a task that is similar or relevant to the original task that the target paper is solving. To create an automatically labelled dataset, we iterate over all sentences in the research corpora. If a sentence contains the words - Related Work or Previous Work or Baseline, then we take 15 sentences occurring after it. We assign positive labels to sentences containing model entity mentions by referring to our KB. We consider the maximal span for entity matching between our unlabelled text and KB. For creating negative samples, we randomly sample from all sentences and make sure the above words are absent and it also does not contain model entity mentions. We keep an equal distribution of positive and negative labels. An example of a positive and negative label is shown in Figure 3.3.

3.9.1 Training the classifier

The most commonly used approach of averaging BERT embeddings or using the output of the first token (the [CLS] token) yields subpar sentence representations [55]. Hence, we choose Sentence-BERT [55], a modification of the pretrained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using dot product. It takes a sentence as input and returns the corresponding sentence-level representation as output. We use Sentence-BERT to encode the sentences and use

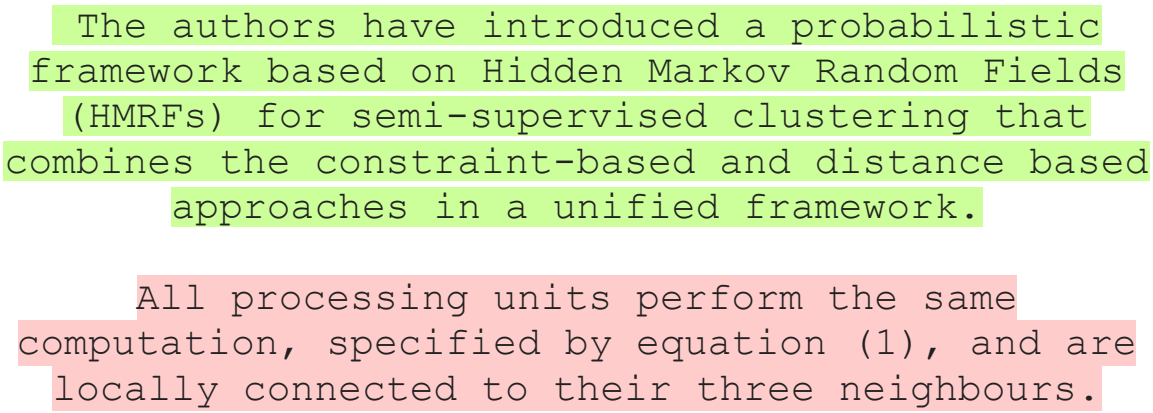

Figure 3.3: Sentence Intent : Positive label labelled with green, negative labelled with red

Logistic Regression as our binary classifier to train it on 15,518 labelled sentences, containing both citation and non-citation sentences. Positive and negative samples are equally distributed. The sentence dataset size is kept small to avoid compromising on the quality of the labels. The train-test split followed is 75-25. The testset accuracy (which, again, consists of both citation and non-citation sentences) is 86.41%.

3.10 Entity Citation Linker

For the entity citation linking, we iterate between all possible extracted entity and citation combination and get their closeness score, which is the string distance between an entity and the citation occurrence. We first take all the citations and keep the closest entity per citation.Then, we take all the entities and keep the closest citations per entity. This linking process is able to accurately link most of the extracted entities with their closest citations, as demonstrated by [19].

3.11 Error Analysis

We conduct error analysis for model entity extraction, sentence intent classification and entity citation linking. Some precision error is introduced into the model because for the

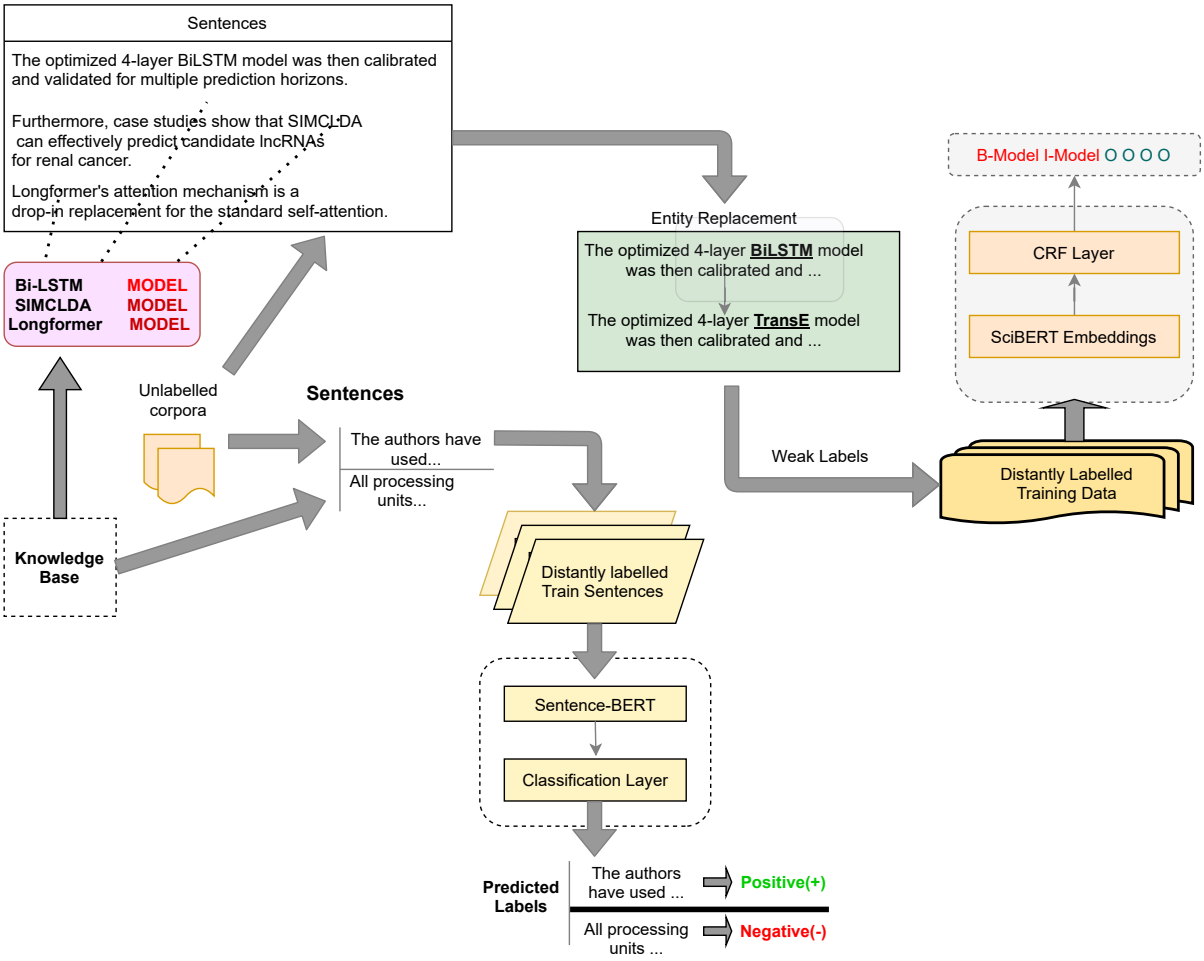

Figure 3.4: Training Pipeline for the Sentence Intent Classifier and NER model using unlabelled corpora and KB

training set we consider maximum span of each entity and the I-Model entity occurrence (a token that lies inside a named entity) is high. We find in our evaluation dataset, the number of B-Model entity is massively more, which leads to the model misclassifying an O as an I for few sentences. Also, due to the usage of citation sentences in the evaluation dataset, our model recognises the citation marker occurring right after the entity as an I-Model. Also, most of the citation sentences in the evaluation dataset has a large number of named entities occurring adjacently, as seen in many citation contexts. The model, which is trained on sentences from abstracts only, is unable to recognise all of them as entities sometimes. For the sentence intent classification, our classifier often recognises sentences containing dataset names as a positive label. This can be attributed to the fact that citation sentences that refer to different datasets often have a similar structure to those citing model names of prior work.Lastly, for the entity citation linker, sometimes an entity that is associated with a citation marker occurs in the

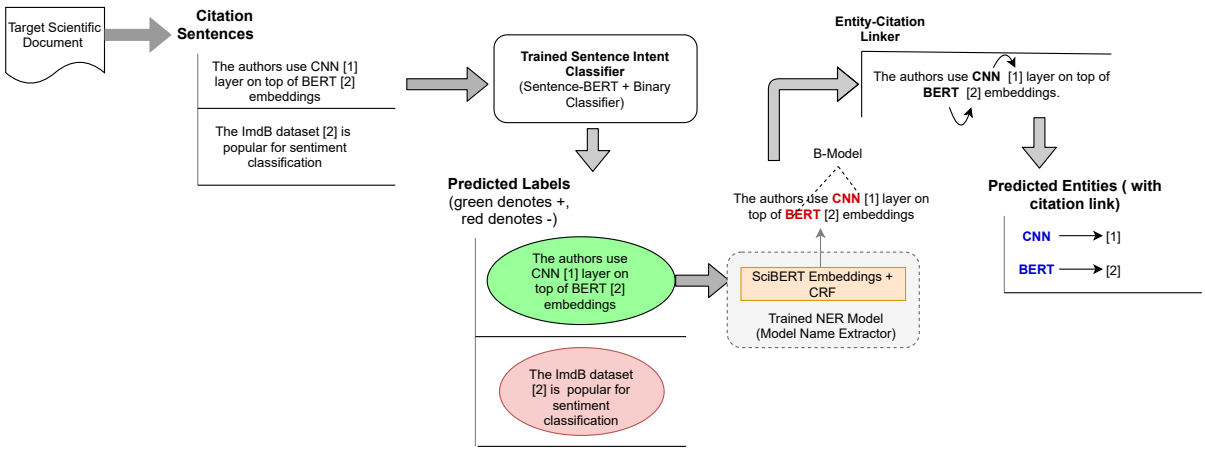

Figure 3.5: Entire Automated Framework utilising the trained models. The input is a scientific document and the output from the pipeline is a set of predicted model entities linked with their citation.

initial part of a sentence and its not the closest to the citation. This can lead to missed out or incorrect linking.

3.12 Implementation details

We use PyTorch framework to implement our NER model. We use the pre-trained SciBERT tokenizer and embeddings as input to a dropout layer with a dropout probability of 0.5 to prevent overfitting. We use learning rate of 1e-5 and train all models for 10 epochs.We pass the output from the dropout layer through a linear layer with input dimension same as the hidden dimension of SciBERT embeddings (768) and output dimension same as the number of labels (4). For Sentence-BERT,we use pretrained models available in Pytorch.

Chapter 4

Transformer Based Architectures for Complex NER in English

4.1 Overview

We investigate the task of complex NER for English language. The task is non-trivial due to the semantic ambiguity of the textual structure and the rarity of occurrence of such entities in the prevalent literature. Using pre-trained language models, we obtain a competitive performance on this task. We qualitatively analyse performance of multiple architectures for this task. All our models are able to outperform the baseline by a significant margin. Our best performing model advances the baseline F1-score by over 9%.

4.2 Introduction

Named Entity Recognition is an Information Extraction task that aims to detect entities from unstructured text and classify them into predefined categories. Although the task of NER has been investigated adequately by previous research work [44, 48, 29, 18, 57], the detection of named entities in a multilingual setting is non-trivial. Furthermore, the introduction of additional layers of complexity - in the form of semantic ambiguity and a lower amount of contextual availibility poses further challenges. NER in low resource languages further enhances the difficulty of such task due to scarcity of available data. Recently, deep learning models have gained popularity for NER [68, 34, 21]. However, these approaches are data-intensive and become ineffective when there is a lack of labelled data. Hence, the NER task for low-resource languages becomes further challenging.

To foster research in this area, the SemEval MultiCoNER challenge [43] was introduced that deals with multiple low-resource language NER with semantically ambiguous entities. In this paper, we describe our approach to tackle this task using state-of-the-art deep learning models and introduce a simple neural network architecture that builds on top of pre-trained language models. Our approach beats the baseline by a significant margin. We compare multiple architectures on the test and validation set of the shared task. All our models beat

the baseline by a significant margin. We provide the formal task description in Section 5.3, the dataset details in Section 5.5, the method and the model architecture in Section 4.5. We provide details about the experimental implementation in Section 5.6. We discuss the results obtained and error analysis in Sections 5.7 and 5.8 respectively.

4.3 Task Description

The objective of this shared task is to build complex Named Entity Recognition systems for multiple languages such as English, Spanish, Chinese, Hindi, Bangla, etc. The task presents a unique challenge in the form of detecting the entities in semantically ambiguous and low-context settings. Moreover, the shared task also tests the generalization capability and domain adaptability of the proposed systems by testing the system over additional (low-context) data sets containing questions and short search queries, such as Google Search queries.

For this task, the systems had to identify the B-I-O format [54] (short for beginning, inside, outside) tags for six NER-tags classes, namely Person, Product, Location, Group, Corporation, AND Creative Work.

Earlier works have also tried to address the problem of NER, but usually, the datasets consisted of well-formed texts of easy entities [2], and little has been done to tackle the problem of identifying semantically and syntactically ambiguous entities like Creative Works. For example : *Eternal Sunshine of the Spotless Mind* and *Among Us* are complex entities, that may be considered as Named Entities in some very selective contexts for eg. *Among Us* is not a NE in "There is not much disagreement among us", but a CW in " *Among Us* is a super fun game to play". This task also aims at tackling such problems.

4.4 Dataset

The MultiCoNER dataset [42] introduced consists of labelled complex Named Entities. For the monolingual track, the participants have to train a model that works for one language only. For training and validation purposes, train and dev set is provided with labelled entities. The monolingual model trained needs to be used for the prediction of named entities in the test set. The labels from the test set are not provided directly. In this system description for the monolingual track, we have considered the English NER dataset for our task. The dataset follows a BIO tagging scheme and there are 6 entity types in the label space. The statistics for the English dataset in the monolingual track for the train and dev set are provided in Table 5.1 and the description of the label space in Table 5.2.

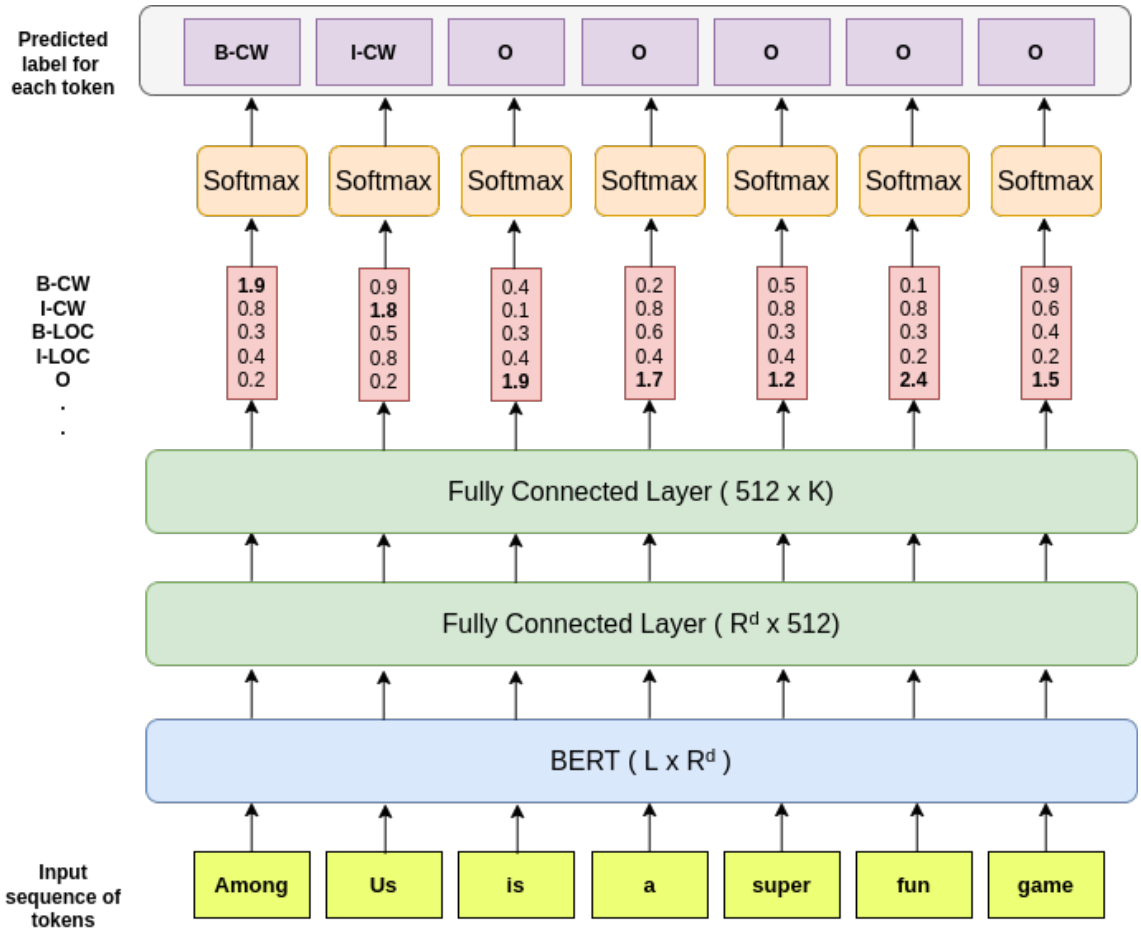

Figure 4.1: BERT-CRF architecture

4.5 Models

This section describes our approach to designing a system to solve the problem of classifying the tokens of a given sentence into one of the six NE categories. We also briefly describe the BERT model architecture employed in our system.

As is the case with most of the NLP tasks, the performance of the model boils down to learning the best-distributed representation for the tokens. With the advent of transformer-based models, the whole domain of NLP has been revolutionized because they provide us with some of the most feature-rich embeddings. Contextual embeddings learned using transformer-based models give better performance than embeddings learned using traditional methods such as TF-IDF, word2vec, etc., for downstream tasks such as NER, since such tasks require greater contextual awareness.

We also adopted the simple strategy of finetuning various architectures based on pre-trained language models such as Bert on our task-specific data.

BERT+CRF : We use a pretrained BERT model to obtain the token embeddings. These embeddings are passed to a token-level classifier followed by a Linear-Chain CRF. The CRF produces probability distribution over the entire label space for each token among the sequence of tokens. More formally: 1) For a sequence of tokens $x = (x_1, x_2, x_3, ..., x_m)$, where x_i is the ith token among the sequence of tokens, we obtain a low-dimensional dense embedding, $x_i \in R^d$ where d is the embedding dimension. 2) This embedding is mapped to a lower dimensional space $x_i \in R^k$ where k is the total number of labels. 3) The output scores from the linear layer are obtained as $P \in R^{m \times k}$, where m is the number of tokens. These scores are passed to the CRF layer, whose parameters are $A \in R^{k+2 \times k+2}$. Each element A_{ij} signifies the transition score from the ith label to the jth label. The 2 additional states in A are the start and the end state of a sequence. For a series of tokens $x = (x_1, x_2, x_3, ..., x_m)$ we obtain a series of predictions $y = (y_1, y_2, y_3, ..., y_m)$. As described in [30], the score of the entire sequence is defined as :

$$s(x, y) = \sum_{i=0}^{m} A_{y_i, y_{i+1}} + \sum_{i=1}^{m} P_{i, y_i}$$

The model is trained to maximise the log probability of the correct label sequence:

$$\log(p(y|x)) = s(x, y) - \log\left(\sum_{\tilde{y} \in Y_X} e^{s(x, \tilde{y})}\right)$$

where Y_X are all possible label sequences.

BERT+BiLSTM+CRF : We use a pretrained BERT model to obtain the contextual embeddings from the sentences. These embeddings are passed to the BiLSTM layer. The BiLSTM layer captures these into a hidden state representation. This representation is passed to a CRF layer that obtains the probability distributions across the labels. Specifically, the pretrained language model is used to map the tokens in each sentence to a distributed representation. This is used as the word embedding layer of the BiLSTM-CRF model. The BiLSTM-CRF layer is used to sequence label the sentence, and the predicted labels are obtained. The supervised learning algorithm iterates to improve its predicted label accuracy over every iteration. More formally, the process can be described as follows : 1) The target sentence comprising of m tokens, is represented as $x = (x_1, x_2, x_3, ..., x_m)$, where x_i represents the ith token of the entire target sentence. 2) x_i is mapped to a low dimensional dense vector, $x_i \in R^d$ using the pretrained BERT embeddings, where d is the dimension of dense embedding. 3) The sequence of tokens x is taken as an input to the BiLSTM in each time step, and the forward hidden states $\overrightarrow{h_f} = (\overrightarrow{h_1}, \overrightarrow{h_2}, \overrightarrow{h_3}, ..., \overrightarrow{h_m})$ and the backward hidden states $\overleftarrow{h_b} = (\overleftarrow{h_1}, \overleftarrow{h_2}, \overleftarrow{h_3}, ..., \overleftarrow{h_m})$ are concatenated to form the combined hidden state representation $h = [\overrightarrow{h_f}, \overleftarrow{h_b}]$. 4) The combined hidden state representation $h \in R^{m \times n}$ is reduced to a k dimensions using a linear layer, where k is the number of labels to distribute the probabilities across. 4) Finally, the CRF layer is used to obtain the probability distribution across all the labels to obtain the final prediction.

	Train	**Dev**
# sentences	15300	800

Table 4.1: Total sentences in English monolingual track

Label	**Description**
PER	Person
LOC	Location
GRP	Group
CORP	Corporation
PROD	Product
CW	Creative Work

Table 4.2: Entity Types in the label space

	BERT+Linear			**BERT+CRF**			**BERT+BiLSTM-CRF**		
Class Label	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
LOC	0.9304	0.9145	**0.9224**	0.903	0.9145	0.9087	0.9025	0.9103	0.9064
PER	0.9659	0.9759	**0.9708**	0.936	0.9586	0.9472	0.8882	0.9586	0.9221
PROD	0.7365	0.8367	0.7834	0.7785	0.7891	**0.7838**	0.7372	0.7823	0.7591
GRP	0.8923	0.9158	**0.9039**	0.8341	0.9000	0.8658	0.8466	0.8421	0.8443
CW	0.7955	0.7955	**0.7955**	0.7963	0.733	0.7633	0.7353	0.7102	0.7225
CORP	0.893	0.8653	**0.8789**	0.8877	0.8601	0.8737	0.8837	0.7876	0.8329
Average	0.8689	0.8839	**0.8758**	0.8559	0.8592	0.8571	0.8322	0.8318	0.8312

Table 4.3: Results of our models on validation dataset

BERT+Linear: The token sequence is mapped to a lower dimensional space using pre-trained BERT embeddings. These embeddings are then passed to a linear layer, that maps these embeddings to a lower dimension of label space. The output scores are then softmaxed to provide a probability distribution across all labels.

4.6 Implementation Details

We implement all our transformer based models using Pytorch and Huggingface library. We implemented 3 models, BERT+Linear, BERT+CRF and BERT+BILSTM+CRF. WE also tried adding POS embeddings as extra features to the models and compared the results. We use a dropout from 0.2 to 0.5 in all models, and found that 0.3 gave the best results throughout. We used 2 linear layers in the BERT+Linear model. We added a softmax layer to obtain the probability distribution across all the labels. For the BERT+Linear model, we run our experiments across 1-20 epochs. We find that the model starts to overfit after 10 epochs, and the best results are obtained after 5 epochs of training. For BERT+CRF, we experiment across 1-100 epochs. We find the model gives the most optimal result at the 20th epoch, after which it starts to overfit. We use a learning rate of $1e^{-4}$ for all the models. WE validate the results of all models using our dev set.

4.7 Results

We compare the performance of our models in the validation set against the baseline. We use the best performing model for the final submission in the evaluation phase. We provide details of the performance of the best performing over the blind test dataset provided in the evaluation phase. We provide the detailed comparison of the performance of our models across all the class labels in the validation dataset in Table 5.3. We observe that the simple BERT+Linear model performs the best as compared to other larger models. We attribute this to the limited number of samples in the training dataset. The lack of sufficient number of training samples limits the ability of larger models to generalise properly over the entire training set. We notice that the performance of BERT+Linear model is consistent across all class labels except for PRODUCT.

	Precision	**Recall**	**F1-Score**
Baseline System	0.773	0.780	0.776
BERT + CRF	0.855	0.859	0.857
BERT+BiLSTM-CRF	0.832	0.831	0.831
BERT + Linear	**0.868**	**0.883**	**0.875**

Table 4.4: Comparison of model performances with baseline on validation dataset

The description attributed to each class label is described in Table 5.2.

	BERT+Linear		
Class Label	Precision	Recall	F1-Score
LOC	0.7292	0.7614	0.7449
PER	0.8776	0.8922	0.8848
PROD	0.7079	0.6460	0.6755
GRP	0.7699	0.6600	0.7107
CW	0.5527	0.6299	0.5888
CORP	0.7253	0.6759	0.6998
Average	0.7271	0.7109	0.7174

Table 4.5: Performance of model on test dataset

4.8 Error Analysis

We perform error analysis for all 3 different model performances on the validation dataset. We find that for all 3 models, each model has the highest difficulty in accurately predicting the *CW (Creating Work)* label. This can be attributed to the higher degree of ambiguity when it comes to *CW* named entities, as these often share similar type of textual structure as normal non-named entity text tokens. It can be inferred that all 3 models are memorising entity names from the training data to some extent. It is most prevalent in **BERT+BiLSTM+CRF** model, as we can see that it has the least amount of prediction accuracy among other models. This is consistent with our reasoning that heavier models tend to overfit the dataset faster. Hence, we deduce that named entity memorisation can be attributed to a type of overfitting behavior by the model in question. The **BERT+Linear** model, which is the lightest model with the least amount of trainable paramenters among all 3, is found to be significantly less prone to memorise entity names.

Furthermore, upon qualitative analysis,we found that our models often have difficulty in recognising longer named entities (entities comprising of 5 or more tokens). This can be attributed to the lack of occurrence of such entities in the training dataset. The models are majorly exposed to a shorter set of entity spans, and texts that occur out of BIO tag and are non-named entity. Due to the lack of exposure of the models to adequate training instances of longer spans, the models are often unable to predict such longer entity spans.

Chapter 5

Complex NER in Semantically Ambiguous Settings for Low Resource Languages

5.1 Overview

We leverage pre-trained language models to solve the task of complex NER on 2 low-resource languages- Chinese and Spanish. We use the technique of Whole Word Masking (WWM) to boost the performance of Masked Language Modeling objective on large, unsupervised corpora. We experiment with multiple neural network architectures, incorporating CRF, BiLSTMs and Linear Classifiers on top of pre-trained BERT embeddings. All our models outperform the baseline by a significant margin and our best performing model obtains a competitive position on the evaluation leaderboard for the blind test set. We hope this work facilitates further research in the challenging domain of ambiguous, low-resource, complex NER.

5.2 Introduction

We investigate the task of complex, semantically ambiguous, and low-resource NER [43]. The most popular NER task in the English language is CoNLL [4], which is widely used as a benchmark for most NER models. Multiple models have been able to obtain sufficiently high performances in this task setting [63, 74, 39, 58, 72, 69, 64]. The CoNLL training-set consists of 14,987 train sentences comprising of 203,621 tokens for English data. The entity space consists of 4 different types of entity type labels (locations, persons, organisations and miscellaneous) to classify each named token. The English data was taken from the Reuters Corpus, which comprises of Reuters News Stories for 1 year. The training data source, and by extension, the labelled named entities comprises of majorly popular entities found in the general English textual content prevalent in the media. Hence, these entities were easier to classify into the correct tokens due to the large prevalence of training data. With the use of pretrained transformer based language models which are already trained on a large unlabelled corpora of

English text, this task became even less challenging, as the nature of textual structure in these corpora largely overlap with that of CoNLL.

However, there is a multitude of varieties of named entities possible, ones that comprise of complex, ambiguous textual structural content. Such named entities are harder in general to predict for language models, due to the semantically ambiguous nature of the textual structure of the named entities, and the lower amount of occurrence of such entity types in general English text. The shared task of MultiCoNER (stands for multilingual, complex NER) adds additional challenge by introducing rarer label types (like creative work,product etc).

Another way to increase the difficulty of NER tasks is to perform it for low-resource languages. There is a significant dearth of both labelled and unlabelled data for such languages. The complexity is further enhanced by using rarer entity types in such languages. Combined with a lack of unlabelled data, the lack of occurrence of rarer entity token types becomes even harder for the fine-tuned language models to overcome. The shared task of MultiCoNER introduces datasets in multiple low-resource languages.

We leverage large pretrained language models trained in low-resource language corpora to obtain competitive performances in the low-resource, complex NER setting. We show that simpler architectures successfully outperform other heavier counterparts. We use standard BERT-CRF based models to obtain high performances in the evaluation set. We experiment on two low-resource dataset: Spanish and Chinese.

Our approach beats the baseline by a significant margin. We compare multiple architectures on the test and validation set of the shared task. All our models beat the baseline by a significant margin. We provide the formal task description in Section 5.3, the dataset details in Section 5.5, the method and the model architecture in Section 4.5. We provide details about the experimental implementation in Section 5.6. We discuss the results obtained and error analysis in Sections 5.7 and 5.8 respectively.

5.3 Task Description

The objective of this shared task is to build complex Named Entity Recognition systems for multiple languages such as English, Spanish, Chinese, Hindi, Bangla, etc. The task presents a unique challenge in the form of detecting the entities in semantically ambiguous and low-context settings. Moreover, the shared task also tests the generalization capability and domain adaptability of the proposed systems by testing the system over additional (low-context) data sets containing questions and short search queries, such as Google Search queries.

For this task, the systems had to identify the B-I-O format [54] (short for beginning, inside, outside) tags for six NER-tags classes, namely Person, Product, Location, Group, Corporation, AND Creative Work.

Earlier works have also tried to address the problem of NER, but usually, the datasets consisted of well-formed texts of easy entities [2], and little has been done to tackle the problem of identifying semantically and syntactically ambiguous entities like Creative Works. For example : *Eternal Sunshine of the Spotless Mind* and *Among Us* are complex entities, that may be considered as Named Entities in some very selective contexts for eg. *Among Us* is not a NE in "There is not much disagreement among us", but a CW in " *Among Us* is a super fun game to play". This task also aims at tackling such problems.

5.4 Dataset

The MultiCoNER dataset [42] consists of multiple low-resource languages. We consider Chinese and Spanish langauge in this paper. For the monolingual track, the participants have to train a model that works for one language only. We train the finetune the language model in the train set to obtain predictions in dev and test set. The labels from the blind test set are not provided directly. The dataset follows a BIO tagging scheme and there are 6 entity types in the label space. The statistics for the Chinese and Spanish dataset in the monolingual track for the train and dev set are provided in Table 5.1 and the description of the label space in Table 5.2.

5.5 System Overview

At first we pre-train the BERT language model on unlabelled corpora for the target low resource language. For Chinese, we use the strategy outlined by [12]. BERT uses the WordPiece tokenizer [66] to split tokens into smaller fragments. It is easier for the Masked Language Model to predict these masked fragments. However, for the Chinese textual texture, the Chinese characters are not formed by alphabet-like symbols, so the WordPiece tokenizer is unable to split the words into small fragments. Hence, we use the Chinese Word Segmentation (CWS) tool to split the text into separate words, and then use Whole Word Masking strategy for the Masked Language Model objective. This removes the drawback of masking small fragments, making it harder for the model to predict whole masked words.

For the spanish variant, we adopt the strategy outlined by [10]. Similar to [12], they use the strategy of whole word masking for pre-training BERT language model on unlabelled Spanish corpus.

We adopt the strategy of finetuning these pre-trained BERT models on the downstream NER task for each language respectively.

BERT+CRF: We obtain token-level dense representations using BERT-based pretrained embeddings. We pass these embeddings to the CRF layer to obtain the probabaility distribution across the label space. For a sequence of tokens $x = (x_1, x_2, x_3, ..., x_n)$, we obtain the ith token

representation x_i of dimension d, which is the dimension of the dense vector representations of the BERT-based embeddings obtained from the pre-trained language model. The token emedding x_i is passed to a dense linear layer to transform the representation from d to k dimensional space, where k is the number of labels. The output scores, obtained from the linear layer as $P \in R^{m \times k}$, are passed to the CRF layer whose parameters are $A \in R^{k+2 \times k+2}$. Element A_{ij} denotes the transition score from the ith to the jth label. 2 additional states are added to the start and end of the sequence. For a series of tokens $x = (x_1, x_2, x_3, ..., x_n)$ we obtain a series of predictions $y = (y_1, y_2, y_3, ..., y_n)$.

As described in [30], the score of the entire sequence is defined as :

$$s(x, y) = \sum_{i=0}^{m} A_{y_i, y_{i+1}} + \sum_{i=1}^{m} P_{i, y_i}$$

The model is trained to maximise the log probability of the correct label sequence:

$$\log(p(y|x)) = s(x, y) - \log\left(\sum_{\tilde{y} \in Y_X} e^{s(x, \tilde{y})}\right)$$

where Y_X are all possible label sequences.

BERT+BiLSTM+CRF : We obtain token-level contextual dense representations using BERT-based pretrained embeddings. These embeddings are passed to a BiLSTM layer which obtaines the hidden-state representation of these tokens. We pass these hidden states to the CRF layer to obtain the probabaility distribution across the label space. We use the pre-trained language model to map the tokens in each sentence to a dense embedding representation. The BERT-based dense embeddings are passed to the BiLSTM-CRF layer, which is used to obtain the predicted labels for each tokens in the entire sequence. More formally, For a sequence of tokens $x = (x_1, x_2, x_3, ..., x_n)$, we obtain the ith token representation x_i of dimension d, which is the dimension of the dense vector representations of the BERT-based embeddings obtained from the pre-trained language model. The token emedding x_i is passed to a dense linear layer to transform the representation from d to k dimensional space, where k is the number of labels. The sequence of tokens x is taken as an input to the BiLSTM in each time step, and the forward hidden states $\overrightarrow{h_f} = (\overrightarrow{h_1}, \overrightarrow{h_2}, \overrightarrow{h_3}, ..., \overrightarrow{h_n})$ and the backward hidden states $\overleftarrow{h_b} = (\overleftarrow{h_1}, \overleftarrow{h_2}, \overleftarrow{h_3}, ..., \overleftarrow{h_n})$ are concatenated to form the combined hidden state representation $h = [\overrightarrow{h_f}, \overleftarrow{h_b}]$. The combined hidden state representation $h \in R^{m \times n}$ is transformed to a k dimensional space using a linear layer, where k is the number of labels to distribute the probabilities across. Finally, the CRF layer outputs the probability distribution for each token across the label space.

BERT+Linear: The token sequence is mapped to a lower dimensional space using pre-trained BERT embeddings. These embeddings are then passed to a linear layer, that maps these embeddings to a lower dimension of label space. The output scores are then softmaxed to provide a probability distribution across all labels.

	Train	**Dev**
# sentences	15300	800

Table 5.1: Total sentences in Chinese and Spanish monolingual track

Label	**Description**
PER	Person
LOC	Location
GRP	Group
CORP	Corporation
PROD	Product
CW	Creative Work

Table 5.2: Entity Types in the label space

	BERT+CRF			**BERT+Linear**			**BERT+BiLSTM-CRF**		
Class Label	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
LOC	0.8368	0.8796	**0.8577**	0.8194	0.8613	0.8399	0.8219	0.8759	0.8481
PER	0.9065	0.9028	0.9047	0.8933	0.9150	0.9040	0.9177	0.9028	**0.9102**
PROD	0.6970	0.7468	0.7210	0.6864	0.7532	0.7183	0.7278	0.7468	**0.7372**
GRP	0.7952	0.7857	0.7904	0.8061	0.7917	**0.7988**	0.7751	0.7798	0.7774
CW	0.7965	0.7135	0.7527	0.8107	0.7135	**0.7590**	0.7654	0.7135	0.7385
CORP	0.8657	0.8227	**0.8436**	0.8529	0.8227	0.8375	0.8397	0.7801	0.8088
Average	0.8163	0.8085	**0.8117**	0.8115	0.8096	0.8096	0.8079	0.7998	0.8034

Table 5.3: Results of our models on validation dataset for Spanish language

	BERT+CRF			**BERT+Linear**			**BERT+BiLSTM-CRF**		
Class Label	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
LOC	0.9239	0.9312	0.9275	0.9186	0.9259	0.9223	0.9465	0.9365	**0.9415**
PER	0.8971	0.9457	**0.9208**	0.8955	0.9302	0.9125	0.8497	0.9225	0.9084
PROD	0.8662	0.8504	**0.8582**	0.8593	0.8248	0.8417	0.8867	0.8285	0.8566
GRP	0.7727	0.6538	0.7083	0.6923	0.6923	0.6923	0.7500	0.6923	**0.7200**
CW	0.8556	0.8191	0.8370	0.8370	0.8191	0.8280	0.8265	0.8617	**0.8437**
CORP	0.8808	0.8854	**0.8831**	0.8883	0.8698	0.8789	0.8615	0.8750	0.8682
Average	0.8660	0.8476	0.8558	0.8485	0.8437	0.846	0.8610	0.8527	**0.8564**

Table 5.4: Results of our models on validation dataset for Chinese language

5.6 Implementation Details

We implement all our transformer based models using Pytorch and Huggingface library. The Chinese language model with the Whole Word Masking (WWM) objective is trained on the Chinese wikipedia unlabelled text corpus. We use the same training corpus as [10] to pre-train the BERT language model on Spanish data. We implement 3 models : BERT-BiLSTM-CRF, BERT+Linear, BERT+CRF for our low resource NER task setting. We run our experiments between 1-100 epochs. WE find that the best results are obtained after 10 epochs of training for each model after which the model starts to overfit. We use a cyclic learning rate between $1e^{-4}$ to $1e^{-6}$. We use a dropout from 0.2 to 0.5 for all models. We validate the results of all models using our validation dataset.

5.7 Results

We compare the performances of all models in the low-resource language setting for both langauges. We observe that the **BERT+CRF** model performs the best across both langauges. We choose the best performing model to evaluate our results on the blind test set. Our approach beats the baseline by a significant margin and outperforms multiple models in the competition. We provide detailed comparisons of all 3 models in the Tables 5.3 and 5.4 for Spanish and Chinese language respectively. We also compare the results between the baseline and our models for the validation dataset, in the Tables 5.5 and 5.6.

We observe the **BERT+CRF** model beats **BERT+Linear** by a slender margin. This can be attributed to the addition of the CRF layer, which has been popularly used for sequence labelling tasks by various neural architectures. The **BERT+BiLSTM+CRF** model is much

	BERT+CRF		
Class Label	Prec	Rec	F1
LOC	0.5768	0.6571	0.6144
PER	0.7641	0.7739	0.7690
PROD	0.6292	0.5141	0.5659
GRP	0.5727	0.5560	0.5642
CW	0.5331	0.5257	0.5294
CORP	0.6605	0.6005	0.6291
Average	0.6227	0.6046	0.6120

Table 5.5: Performance of spanish model on test dataset

heavier with a larger number of parameters, and overfits the training dataset due to the smaller number of training instances.

5.8 Error Analysis

We perform error analysis on all 3 different models. We qualitatively analyse the predictions on the validation dataset for both languages. As the final evaluation test set in blind, we are unable to perform analysis on the same.

We find that the labels GRP (Group), PROD (Product) and CW (Creative Work) are the most inaccurately predicted labels for the Spanish models. This conforms to our hypothesis that the long-tailed nature of these entities (which means, the frequency of occurrence of such entity types in the general literature of the target language is rare). Hence, the model has the most difficulty in recognising these entities from the contextual sentences. The other label types are more common and was present in the CoNLL dataset as well. We also notice that the **BERT+Linear** does marginally better than **BERT+CRF** on predicting such labels, despite it not being the best performing model overall. This can be attributed to it being a lighter model, imparting it the capability of generalising better while training on a relatively lower amount of training instances. The other 2 models have a larger number of parameters, leading them to overfit due to label scarcity. For the Chinese language as well, we notice a similar phenomenon, for GRP label.

We perform error analysis for all 3 different model performances on the validation dataset. We find that for all 3 models, each model has the highest difficulty in accurately predicting the *CW (Creating Work)* label. This can be attributed to the higher degree of ambiguity when it comes to *CW* named entities, as these often share similar type of textual structure as normal

	BERT+CRF		
Class Label	Prec	Rec	F1
LOC	0.6930	0.7955	0.7407
PER	0.7952	0.6377	0.7078
PROD	0.6853	0.7232	0.7038
GRP	0.7254	0.4608	0.5636
CW	0.5520	0.6798	0.6093
CORP	0.6526	0.7361	0.6918
Average	0.6839	0.6722	0.6695

Table 5.6: Performance of chinese model on test dataset

non-named entity text tokens. It can be inferred that all 3 models are memorising entity names from the training data to some extent. It is most prevalent in **BERT+BiLSTM+CRF** model, as we can see that it has the least amount of prediction accuracy among other models. This is consistent with our reasoning that heavier models tend to overfit the dataset faster. Hence, we deduce that named entity memorisation can be attributed to a type of overfitting behavior by the model in question. The **BERT+Linear** model, which is the lightest model with the least amount of trainable paramenters among all 3, is found to be significantly less prone to memorise entity names.

Furthermore, upon qualitative analysis,we found that our models often have difficulty in recognising longer named entities (entities comprising of 5 or more tokens). This can be attributed to the lack of occurrence of such entities in the training dataset. The models are majorly exposed to a shorter set of entity spans, and texts that occur out of BIO tag and are non-named entity. Due to the lack of exposure of the models to adequate training instances of longer spans, the models are often unable to predict such longer entity spans.

Chapter 6

Conclusion and Future Work

In Chapter 3, we have introduced a novel task of long-tailed model entity recognition from scientific documents. We test our gold standard evaluation set on multiple baselines. We also find a simple strategy of entity replacement works well on small labelled datasets for distant supervision. We integrate our model in the automated pipeline framework. For future work, we aim to utilise this pipeline on a large research corpus to obtain a map of benchmarked model names linked with their respective papers on a massive scale. We believe our work will serve as a starting point for mapping the research landscape of computer science.

In Chapter 4, we have experimented with 3 model architectures for a novel dataset introduced for the shared task of detection of complex NER. Our best performing model comprises of a simple linear classifier on top of BERT based pretrained langugage model. We find that this simple approach performs competitively as compared to its heavier counterparts. It also beats numerous teams in the performance in the final evaluation dataset. Upon analysis, we attribute this observation to scarcity of labelled training data. We find this simpler approach to give a higher performance as it is able to utilise the contextual information from a sequence of tokens to accurately predict the named entity tokens. It is able to optimally avoid overfitting to a larger extent and hence performs better than other heavier models. For future work, we aim to utilise other data augmentation techniques and distant supervision to create clean silver labels in order to increase our training instances. We believe that this would help us leverage larger models for training purposes.

In Chapter 4, we have introduced strong improvements over the baseline for the shared task of complex NER for low resource languages. We leverage the Whole Word Masking objective to obtain a better performance in this low resource setting. We perform extensive experiments and find that simple BERT-CRF based models performs strongly against other heavier models even in such low resource semantically ambiguous setting as evident by the final evaluation rankings. We also conduct qualitative error analysis and describe our findings. For future work, we aim to leverage data augmentation and distant supervision techniques to circumvent the label scarcity problem in low resource languages.

Related Publications

- Swayatta Daw and Vikram Pudi. **Extraction of Competing Models using Distant Supervision and Graph Ranking**. In Scientific Document Understanding at the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22).
- Swayatta Daw and Vikram Pudi. **Long Tailed Entity Extraction of Model Names using Distant Supervision**. In Bibliometric-enhanced Information Retrieval at the 44th European Conference on Information Retrieval.
- Amit Pandey[1], Swayatta Daw[1], Narendra Unnam, Vikram Pudi **Complex NER in Semantically Ambiguous Settings for Low Resource Languages** In Proceedings of 16th SemEval at the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics
- Amit Pandey[1], Swayatta Daw[1], Vikram Pudi **Transformer Based Architecture for Complex NER** In Proceedings of 16th SemEval at the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics

[1]Equal Contribution

Bibliography

[1] S. Ashwini and J. D. Choi. Targetable named entity recognition in social media. *CoRR*, abs/1408.0782, 2014.

[2] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*, 2017.

[3] I. Augenstein, L. Derczynski, and K. Bontcheva. Generalisation in named entity recognition: A quantitative analysis. *CoRR*, abs/1701.02877, 2017.

[4] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli. Cloze-driven pretraining of self-attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5360–5369, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[5] M. Banko and R. C. Moore. Part-of-speech tagging in context. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 556–561, 2004.

[6] M. S. Bari, S. R. Joty, and P. Jwalapuram. Zero-resource cross-lingual named entity recognition. *CoRR*, abs/1911.09812, 2019.

[7] I. Beltagy, K. Lo, and A. Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[8] K. Bennani-Smires, C. C. Musat, M. Jaggi, A. Hossmann, and M. Baeriswyl. Embedrank: Unsupervised keyphrase extraction using sentence embeddings. *ArXiv*, abs/1801.04470, 2018.

[9] F. Boudin. Unsupervised keyphrase extraction with multipartite graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 667–672, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[10] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*, 2020.

[11] H. L. Chieu and H. Ng. Named entity recognition with a maximum entropy approach. In *CoNLL*, 2003.

[12] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online, Nov. 2020. Association for Computational Linguistics.

[13] S. Danesh, T. Sumner, and J. H. Martin. SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 117–126, Denver, Colorado, June 2015. Association for Computational Linguistics.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[15] C. Dozier, R. Kondadadi, M. Light, A. Vachher, S. Veeramachaneni, and R. Wudali. Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*, pages 27–43. Springer, 2010.

[16] X. Feng, X. Feng, B. Qin, Z. Feng, and T. Liu. Improving low resource named entity recognition using cross-lingual knowledge transfer. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 40714077. AAAI Press, 2018.

[17] C. Florescu and C. Caragea. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[18] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 168–171, 2003.

[19] S. Ganguly and V. Pudi. Competing algorithm detection from research papers. In *Proceedings of the 3rd IKDD Conference on Data Science, 2016*, CODS '16, New York, NY, USA, 2016. Association for Computing Machinery.

[20] S. Goldberg, D. Z. Wang, and C. Grant. A probabilistically integrated system for crowd-assisted text labeling and extraction. *J. Data and Information Quality*, 8(2), Feb. 2017.

[21] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[22] K. S. Hasan and V. Ng. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273, 2014.

[23] M. A. Hedderich, L. Lange, and D. Klakow. ANEA: distant supervision for low-resource named entity recognition. *CoRR*, abs/2102.13129, 2021.

[24] Z. Huang, W. Xu, and K. Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[25] S. A. Jadhav. Detecting potential topics in news using bert, CRF and wikipedia. *CoRR*, abs/2002.11402, 2020.

[26] S. Jain, M. van Zuylen, H. Hajishirzi, and I. Beltagy. Scirex: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, jul 2020.

[27] V. Kocaman and D. Talby. Biomedical named entity recognition at scale. *CoRR*, abs/2011.06315, 2020.

[28] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[29] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[30] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[31] R. Leaman and G. Gonzalez. Banner: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008*, pages 652–663. World Scientific, 2008.

[32] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*, 2017.

[33] J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *ArXiv*, abs/1812.09449, 2018.

[34] J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.

[35] C. Liang, Y. Yu, H. Jiang, S. Er, R. Wang, T. Zhao, and C. Zhang. BOND: bert-assisted open-domain named entity recognition with distant supervision. *CoRR*, abs/2006.15509, 2020.

[36] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.

[37] Q. Liu, P. cheng Li, W. Lu, and Q. Cheng. Long-tail dataset entity recognition based on data augmentation. In *EEKE@JCDL*, 2020.

[38] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[39] J. Luoma and S. Pyysalo. Exploring cross-sentence contexts for named entity recognition with BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 904–914, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics.

[40] X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[41] S. Malmasi, A. Fang, B. Fetahu, S. Kar, and O. Rokhlenko. MultiCoNER: A large-scale multilingual dataset for complex named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809, Gyeongju, Republic of Korea, Oct. 2022. International Committee on Computational Linguistics.

[42] S. Malmasi, A. Fang, B. Fetahu, S. Kar, and O. Rokhlenko. MultiCoNER: a Large-scale Multilingual dataset for Complex Named Entity Recognition. 2022.

[43] S. Malmasi, A. Fang, B. Fetahu, S. Kar, and O. Rokhlenko. SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics, 2022.

[44] A. Mansouri, L. S. Affendey, and A. Mamat. Named entity recognition approaches. *International Journal of Computer Science and Network Security*, 8(2):339–344, 2008.

[45] T. Meng, A. Fang, O. Rokhlenko, and S. Malmasi. GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512, Online, June 2021. Association for Computational Linguistics.

[46] S. Mesbah, C. Lofi, M. V. Torre, A. Bozzon, and G.-J. Houben. Tse-ner: An iterative approach for long-tail entity extraction in scientific publications. In *International Semantic Web Conference*, pages 127–143. Springer, 2018.

[47] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *EMNLP*, 2004.

[48] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[49] F. Nooralahzadeh, J. T. Lønning, and L. Øvrelid. Reinforcement-based denoising of distantly supervised NER with partial annotation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 225–233, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[50] S. Pawar, G. K. Palshikar, and P. Bhattacharyya. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*, 2017.

[51] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[52] A. Rahimi, Y. Li, and T. Cohn. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy, July 2019. Association for Computational Linguistics.

[53] A. Rahimi, Y. Li, and T. Cohn. Multilingual ner transfer for low-resource languages. *ArXiv*, abs/1902.00193, 2019.

[54] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer, 1999.

[55] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[56] S. Rijhwani, S. Zhou, G. Neubig, and J. Carbonell. Soft gazetteers for low-resource named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8118–8123, Online, July 2020. Association for Computational Linguistics.

[57] A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1524–1534, 2011.

[58] S. Schweter and A. Akbik. FLERT: document-level features for named entity recognition. *CoRR*, abs/2011.06993, 2020.

[59] R. B. Tchoua, A. Ajith, Z. Hong, L. T. Ward, K. Chard, A. Belikov, D. J. Audus, S. Patel, J. J. d. Pablo, and I. T. Foster. Creating training data for scientific named entity recognition with minimal human effort. In *International Conference on Computational Science*, pages 398–411. Springer, 2019.

[60] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 855860. AAAI Press, 2008.

[61] X. Wang, Y. Guan, Y. Zhang, Q. Li, and J. Han. Pattern-enhanced named entity recognition with distant supervision. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 818–827, 2020.

[62] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu. Automated concatenation of embeddings for structured prediction. *CoRR*, abs/2010.05006, 2020.

[63] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu. Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2643–2660, Online, Aug. 2021. Association for Computational Linguistics.

[64] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu. Improving named entity recognition by external context retrieving and cooperative learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1800–1812, Online, Aug. 2021. Association for Computational Linguistics.

[65] L. Weston, V. Tshitoyan, J. Dagdelen, O. Kononova, A. Trewartha, K. A. Persson, G. Ceder, and A. Jain. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of chemical information and modeling*, 59(9):3692–3702, 2019.

[66] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[67] J. Xie, Z. Yang, G. Neubig, N. A. Smith, and J. Carbonell. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[68] V. Yadav and S. Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

[69] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online, Nov. 2020. Association for Computational Linguistics.

[70] Y. Yang, W. Chen, Z. Li, Z. He, and M. Zhang. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics.

[71] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

[72] D. Ye, Y. Lin, and M. Sun. Pack together: Entity and relation extraction with levitated marker. *CoRR*, abs/2109.06067, 2021.

[73] Y. Yu and V. Ng. Wikirank: Improving keyphrase extraction based on background knowledge. *ArXiv*, abs/1803.09000, 2018.

[74] W. Zhou and M. Chen. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.