DeepType : Natural Language Understanding by Abstraction

Jonathan Raiman

université
PARIS-SACLAY

DeepType: Natural Language Understanding by Abstraction

DeepType: compréhension du langage naturel par l'abstraction

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 : Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique,
Référent : Faculté des sciences d'Orsay

Thèse préparée au **Laboratoire de Recherche en Informatique** (Université Paris-Saclay, LRI) sous la direction de **Johanne Cohen**, directrice de recherche, et **Michèle Sebag**, directrice de recherche.

Thèse soutenue à Paris-Saclay, le 7 Mars 2023, par

Jonathan Raiman

Composition du jury

Anne Vilnat Professeure, Université Paris-Saclay	Présidente
Laure Soulier Maitresse de conference, Sorbonne Université	Examinatrice
Sylvain Lamprier Professeur, Université d'Angers	Examinateur
Emmanuel Morin Professeur, Nantes Université	Rapporteur
Eric Gaussier Professeur, Université Grenoble Alpes	Rapporteur

THESE DE DOCTORAT
NNT : 2023UPASG016

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : DeepType: compréhension du langage naturel par l'abstraction
Mots clés : Langage Naturel ; Apprentissage Profond ; Représentation de Connaissances ; Réseaux Neuronaux

Résumé : La désambiguïsation et l'Entity Linking sont des tâches où le but est de retrouver le sens et l'identité d'un mot ou d'une expression au sein d'un document : par exemple que signifie le mot "France" dans la phrase "La France a battu le Brésil 3-0 dans la finale de 1998" (un pays ? une armée ? une équipe de football ? etc.) ? Cette question est difficile car elle requiert de connaitre le sens des mots dans leur contexte. Et pourtant, l'Entity Linking est une composante clé des moteurs de recherche type Google, pour la traduction automatisée, le fonctionnement de magasins en ligne, des assistants virtuels comme Siri ou Alexa, ou même au fonctionnement de la bourse due à son utilisation dans des systèmes de négociation automatisés.

L'Intelligence Artificielle Neuro-Symbolique est sous-ensemble de l'Intelligence Artificielle (IA) qui est particulièrement pertinent à l'Entity Linking. Le but de ce sous-domaine est de combiner les atouts de l'Intelligence Artificielle Symbolique avec les avancés venant des méthodes basées sur les réseaux neuronaux. Par exemple, un système Neuro-Symbolique permet à un réseau neuronal d'accéder à des informations symbolique sur Internet en lui donnant accès à un explorateur web pour répondre à des questions plutôt qu'en espérant que le réseau neuronal ait mémorisé à l'avance tous les éléments de réponse possible. Cette thèse présente un corpus d'avancé significative pour l'Entity Linking et l'Intelligence Artificielle Neuro-Symbolique.

En premier temps nous élaborons la première étude mesurant la performance humaine en Entity Linking et servant désormais comme référence.

Dans un deuxième temps avons développons DeepType, le premier système basé sur une représentation d'entité qui utilise la hiérarchie des concepts présents dans des ontologies fabriquées par des humains, afin d'entraîner un réseau neuronal profond pour l'Entity Linking. Nous montrons que de remplacer chaque entité par son emplacement dans une hiérarchie simplifié des concepts, plutôt qu'en utilisant directement les entités, donne lieu à une représentation plus compacte et ayant un plafond de performance qui un égale ou supérieure à celui des humains dans notre étude.

Bien que DeepType établisse un nouvel état de l'art, le réseau neuronal entraîné n'est pas à la hauteur de la performance humaine ni celui du plafond potentiel de la représentation choisie. Par conséquence nous avons créé DeepType 2, la première IA surpassant l'humain en Entity Lining. Cette avancée est principalement grace aux interactions de types —une nouvelle façon de representer une entité en observant les relations entre celle-ci et les autres entités dans un document.

Une limitation restante de DeepType 2 est sa dépendence des interactions de types sur une ontologie structurée (e.g. Wikidata), qui peut contenir des erreurs ou être incomplète dans certaines langues ou domaines moins représentés sur Internet. Nous proposons une solution à ce problème dans DeepType 3 en créant une base de Donnée Relationnelle Neuronale (NeRD) : il s'agit d'une technique permettant d'apprendre à une IA comment representer des entités par leur relations avec d'autres entités en se servant de manière équivalente de données structurées ou non-structurées.

A travers ces quatre résultats clés, cette thèse propose une nouvelle référence humaine pour mesurer la performance en Entity Linking, et des algorithmes d'IA qui établissent un nouvel état de l'art, sont les premiers à surpasser l'humain en Entity Linking, et permettant une meilleur croissance et generalization vers des nouveaux domaines d'applications où l'apprentissage dépend sur des relations implicites entre des concepts.

Title : DeepType: Natural Language Understanding by Abstraction
Keywords : Deep Learning ; Neural Nets ; Natural Language ; Knowledge Representation

Abstract : Entity linking is the task of recovering the underlying identity of a word phrase a word in a document : for instance what does the word “France" refer to in “France beat Brazil 3-0 in the 1998 final" (a country ? an army ? a sports team ? a football team ? etc.) ? This is difficult as it requires to understand the meaning of the words in their full context. And yet, Entity Linking is of critical importance for search engines such as Google, translation, online stores, in intelligent assistants such as Siri or Alexa, or even to the stock market through automated trading systems.

Neuro-Symbolic Artificial Intelligence is a subfield of Artificial Intelligence (AI) that is especially relevant to this task. This subfield seeks to combine the the strength of Symbolic Artificial Intelligence with the recent breakthroughs from Neural methods, by for instance enabling an AI to browse the web to answer a question without having to memorize all the facts ahead of time.

This thesis presents a corpus of breakthrough advances for Entity Linking and Neuro-Symbolic Artificial Intelligence. We establish the first benchmark to measure human performance at Entity Linking.

We then develop DeepType, the first system to propose a representation of entities that takes advantage of the hierarchy of concepts in human knowledge bases to train a deep neural network for Entity Linking. We prove that using our simplified concept hierarchies rather than the prior entity-centric approach yields a representation that is more compact and has a performance ceiling that is equal or higher to human accuracy.

Though DeepType sets a new state of the art, the trained neural network is below the performance ceiling and falls short of human performance. We thus created DeepType 2, the first superhuman AI entity linker. This feat was achieved using type interactions —a novel way to characterize entities by studying the relations they have with other entities in a document.

A limitation of DeepType 2 is the reliance of type-interactions on structured knowledge bases such as Wikidata, which are sometimes flawed or unavailable in low-resource languages. We address this in DeepType 3, by creating the Neural Relational Database (NeRD), a method that teaches the AI to characterize entities through their relation with others via structured or unstructured data.

With these four milestone results, this thesis provides a benchmark to measure performance for Entity-Linking, and AI algorithms that outperform the state of the art, are first to achieve superhuman performance, and enable us to scale and generalize to other domains where learning implicit relations between abstract concepts is required.

Synthesis

Le sujet de cette thèse est de permettre à une machine d'apprendre à se servir de connaissances préexistantes sous divers format pour comprendre le language humain. L'arrivée de système tel que ChatGPT ou Siri présente des importantes opportunités pour que l'intelligence artificielle (IA) ait la capacité d'interagir avec des humains, et d'accomplir des tâches utiles. Le principal défaut des algorithmes d'apprentissage utilisés dans la construction de ces systêmes est l'incapacité à interagir ou acceder à des connaissances externes, sous forme de base de données structurées ou non structurés (texte, image, audio, etc.). Nous proposons une solution à ce problème : un corpus d'algorithme permettant à des systemes d'IA basés sur des réseaux de neurones de pouvoir se servir de connaissances externes, ce qui permet à l'IA de rester à jour et d'ameliorer sa performance sur des taches portant sur des connaissances factuelles, comme la désambiguïsation et l'Entity Linking.

La désambiguïsation et l'Entity Linking sont des tâches où le but est de retrouver le sens et l'identité d'un mot ou d'une expression au sein d'un document : par exemple que signifie le mot "France" dans la phrase "La France a battu le Brésil 3-0 dans la finale de 1998" (un pays ? une armée ? une équipe de football ? etc.) ? Cette question est difficile car elle requiert de connaitre le sens des mots dans leur contexte. Et pourtant, l'Entity Linking est une composante clé des moteurs de recherche type Google, pour la traduction automatisée, le fonctionnement de magasins en ligne, des assistants virtuels comme Siri ou Alexa, ou même au fonctionnement de la bourse due à son utilisation dans des systèmes de négociation automatisés.

L'Intelligence Artificielle Neuro-Symbolique est sous-ensemble de l'Intelligence Artificielle (IA) qui est particulièrement pertinent à l'Entity Linking. Le but de ce sous-domaine est de combiner les atouts de l'Intelligence Artificielle Symbolique avec les avancés venant des méthodes basées sur les réseaux neuronaux. Par exemple, un système Neuro-Symbolique permet à un réseau neuronal d'accéder à des informations symbolique sur Internet en lui donnant accès à un explorateur web pour répondre à des questions plutôt qu'en espérant que le réseau neuronal ait mémorisé à l'avance tous les éléments de réponse possible. Cette thèse présente un corpus d'avancé significative pour l'Entity Linking et l'Intelligence Artificielle Neuro-Symbolique.

En premier temps nous élaborons la première étude mesurant la performance humaine en Entity Linking et servant désormais comme référence.

Dans un deuxième temps avons développons DeepType, le premier système basé sur une représentation d'entité qui utilise la hiérarchie des concepts présents dans des ontologies fabriquées par des humains, afin d'entraîner un réseau neuronal profond pour l'Entity Linking. Nous montrons que de remplacer chaque entité par

son emplacement dans une hiérarchie simplifié des concepts, plutôt qu'en utilisant directement les entités, donne lieu à une représentation plus compacte et ayant un plafond de performance qui un égale ou supérieure à celui des humains dans notre étude.

Model		TAC	AIDA
DeepType 3	μ	**97.74**	**97.87**
	σ	$\pm$0.14	$\pm$0.02
Human Oracle		96.86	96.78
DeepType 2		97.48	97.72
DeepType		90.9	94.9
Yang et al. [179]		-	95.9
De Cao et al. [29]		-	93.3
Févry et al. [42]		94.9	96.7-

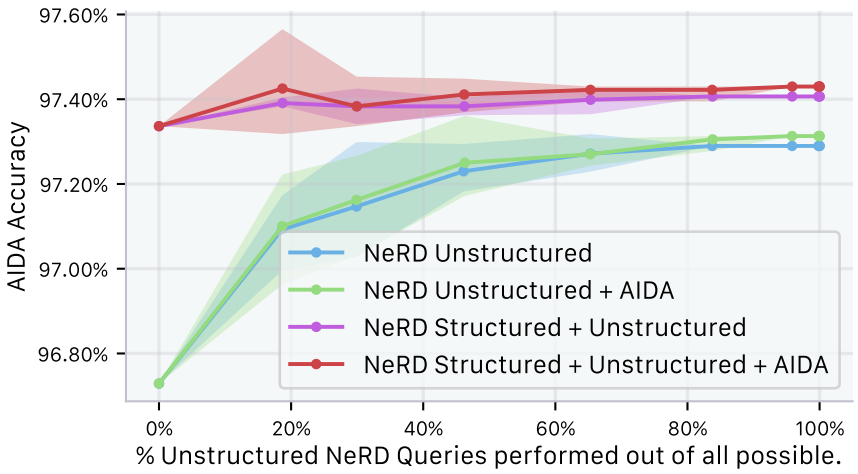
TABLE 1 – L'état de l'art en Entity Linking sur les evaluations TAC and AIDA ($\mu \pm \sigma$, $N = 3$). La plus haute performance est indiquée en gras.

Bien que DeepType établisse un nouvel état de l'art, le réseau neuronal entraîné n'est pas à la hauteur de la performance humaine ni celui du plafond potentiel de la représentation choisie. Par conséquence nous avons créé DeepType 2, la première IA surpassant l'humain en Entity Lining (Table 1). Cette avancée est principalement grace aux interactions de types —une nouvelle façon de representer une entité en observant les relations entre celle-ci et les autres entités dans un document.

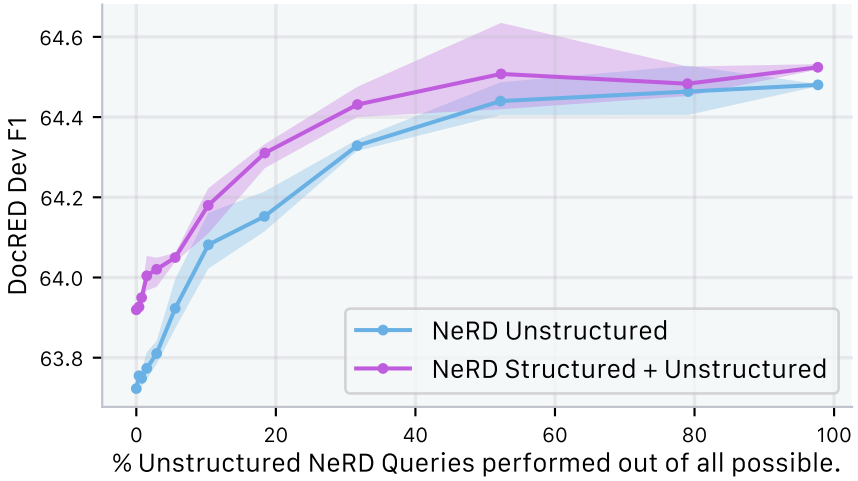
Une limitation restante de DeepType 2 est sa dépendence des interactions de types sur une ontologie structurée (e.g. Wikidata), qui peut contenir des erreurs ou être incomplète dans certaines langues ou domaines moins représentés sur Internet. Nous proposons une solution à ce problème dans DeepType 3 en créant une base de Donnée Relationnelle Neuronale (NeRD) : il s'agit d'une technique permettant d'apprendre à une IA comment representer des entités par leur relations avec d'autres entités en se servant de manière équivalente de données structurées ou non-structurées.

Cette formulation a pour effet de permettre à un systême de beneficier de plusieurs sources d'informations et d'eliminer le travail manuel d'organisation et d'écriture de base de données structurées, ainsi que d'avoir la capacité à ameliorer sa performance en absorbant un corpus de données de plus en plus large (Figure 1).

A travers ces quatre résultats clés, cette thèse propose une nouvelle référence humaine pour mesurer la performance en Entity Linking, et des algorithmes d'IA qui établissent un nouvel état de l'art, sont les premiers à surpasser l'humain en Entity Linking, et permettant une meilleur croissance et generalization vers des nouveaux domaines d'applications où l'apprentissage dépend sur des relations implicites entre des concepts.



(a) Entity Linking avec DeepType 3



(b) Extraction de relation avec RoBERTa-ATLOP + DeepType 3

FIGURE 1 – La performance de DeepType 3's sur AIDA et la precision (F1) de RoBERTa-ATLOP + DeepType 3 sur DocRED augmente avec l'inclusion de plus de données dans la base de donnée. La combinaison de donnnées structurées et non-structurées augmente d'avantage la performance de DeepType 3.

Contents

List of Figures

List of Tables

Introduction

ARTIFICIAL Intelligence (AI) progress has enabled transformative changes to the world around us. This transformation overhauled industries ranging from journalism [12, 170], marketing [16], medecine [168, 73, 92], agriculture [11, 26], transportation [154], to manufacturing [34, 137, 102]. The creative industries are poised to be impacted next, thanks to breakthroughs in AI generated music [33, 149, 56], images [128, 140, 181], video [148, 158, 62], and 3D [46, 117].

Three factors explain recent progress in AI : 1) massive amounts of data, 2) exponential growth of computing power, and 3) new Machine Learning algorithms that leverage the additional data and computation. The explosion of data is linked to the general availability of digital cameras, the growth of the web, crowdsourced encyclopedias such as Wikipedia, and the technological infrastructure underpinning social networks such as Facebook, Youtube, or Twitter. The arrival of programmable graphic processing units (GPUs), alongside the ability to perform datacenter wide computing has also massively increased the amount of computation available to run a Machine Learning experiment [4]. These shifts in the amount of data and computation have led to the developments of Machine Learning algorithms that embrace Richard Sutton's *Bitter Lesson* [153] : general methods tend to outperform specialized ones in the long run. In computer vision and speech recognition, filter banks are replaced by pre-training on large datasets, data augmentations, and deeper neural network architectures capable of discovering better internal representations [122, 20]. Natural language processing has also shifted from engineered features to fine-tuning or querying Pretrained Language Models [19, 23].

Despite these extraordinary achievements, such as super-human performance at object recognition [59], speech recognition [3, 60], language understanding [91], or Go [147], the holy grail of AI, human level performance [17, 98, 50], remains elusive when combining natural language and fine-grained world knowledge.

This milestone remains challenging for several reasons. Firstly, there is insufficient or inadequate data in these specialized knowledge understanding tasks to sufficiently feed present machine learning algorithms. Secondly, the knowledge acquired by existing deep learning machine learning solutions grows stale because they lack access to external resources and human knowledge is usually in symbolic form while neural network architectures used for natural language processing today deal with distributed representations of words and documents. Thirdly, neural network architectures have brittle reasoning due to : a) logical fuzziness caused by operating on distributed representations rather than discrete ones, b) limitations of memory and state representations that degrade with context length.

In practice, the research sub-area of Neuro-Symbolic Artificial Intelligence aims to address these difficulties by combining on the one hand distributed representations and pre-trained neural networks with, on the other hand, an ability access to symbolic resources. Within the Neuro-Symbolic context, this thesis first introduces a human performance benchmark focused on a challenging task for present AI systems, and second presents a strategy to improve the accuracy, sample efficiency,

and transparency of Neuro-Symbolic systems. This strategy increases reasoning robustness, while eliminating the manual effort required to integrate symbolic information in Machine Learning systems, thanks to its ability to leverage existing resources made by humans (Wikidata), including type hierarchies organizing the categories of our concepts.

Neuro-Symbolic Artificial Intelligence

Neuro-Symbolic Artificial Intelligence is a subfield of Artificial Intelligence seeking to combine the strength of Symbolic Artificial Intelligence with the recent breakthroughs from Neural methods [8, 136, 144]. The combination of symbolic and neural methods has become the center of attention in the field of Artificial Intelligence, and its adoption has shown success in areas ranging from graphics to language understanding. Implicit neural models such as NeRFs [100], combine a symbolic model of ray casting combined with a deep neural network and learn how to render an object from any direction. Another family of neuro-symbolic systems combine neural network language models with external tools such as a physics simulators [90] or web browsers [105]. Neuro-Symbolic systems are akin to the emergence of tool use by Artificial Intelligence, and hint at systems that are robust and adaptive. Despite these early successes, several important challenges remain to be solved for Neuro-Symbolic systems to fulfill their promise.

Technical Challenges of Neuro-Symbolic Artificial Intelligence

Neuro-Symbolic Artificial Intelligence faces challenges to train neural networks to manipulate symbols. A first concern, is that direct supervision of Neuro-Symbolic systems will fail due to the non-differentiability of symbolic operations. With sufficient labeled data, it is possible for the neural network to be trained without feedback from symbol manipulation using purely supervised training.

The second challenge of Neuro-Symbolic systems is the interaction with external knowledge in symbolic form while neural network architectures used for processing today deal with distributed representations. There are no general approaches available yet to enable interfacing with external symbols, however natural language based systems using pretrained language models have shown promise at generalizing to new domains and tasks through prompt programming.

The third challenge of Neuro Symbolic systems is that we do not know whether the knowledge organization most useful to humans is also the best one for use by a machine. Indeed, in the “The End of Theory" that Chris Anderson announced [5] there is an assumption that machines can consume data in the same form as humans. One line of research here involves experimenting with different data organizations and understanding which lead to higher accuracy on downstream tasks or are easiest for a machine to learn.

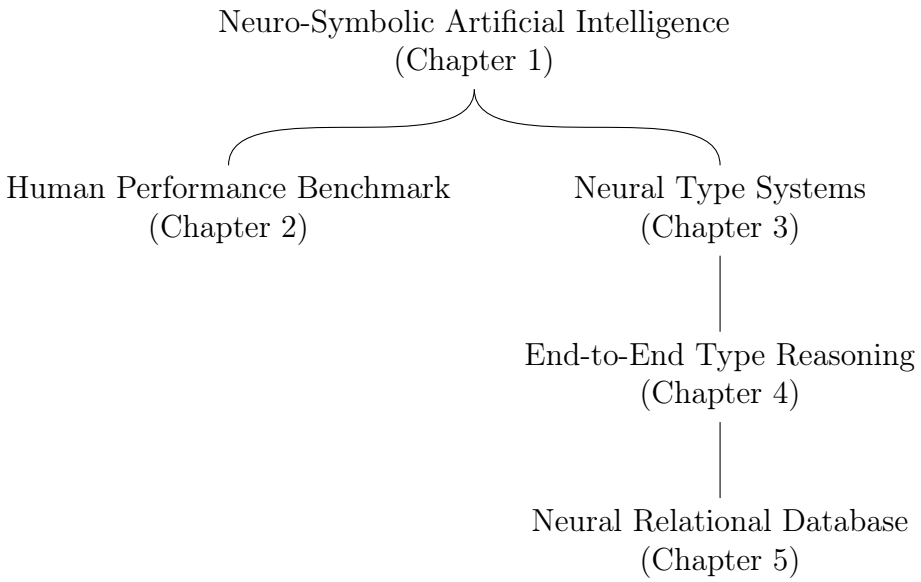


FIGURE 2 – Outline of the thesis

The focus of this thesis is on addressing these three challenges.

Outline of the Thesis

As illustrated in Figure 2, this Thesis presents four contributions addressing issues in the quest for human level performance AI. This thesis is structured as follows :

Chapter I focuses on the formal background of Neuro-Symbolic Machine Learning problem. Starting with the motivation and context of the work, it then presents an overview of the state of the art and related work, followed by an introduction to Neuro-Symbolic Machine Learning (Chapter 1). The introduction next describes the state of the art for the Natural Language Processing tasks studied within this thesis : Named Entity Recognition, Part of Speech Tagging, Entity Linking, and Relation Extraction. Chapters II, III, IV, and V present the four contributions of this thesis (details below). The thesis concludes with a summary and discussion of the contributions, and directions for future work.

Chapter 2 : Human Performance Benchmark We first seek to measure human Entity Linking performance to obtain a meaningful point of comparison for the AI. We establish the first Entity Linking human benchmark that measures human performance and provides a milestone for algorithmic approaches. Specific challenges of this step include ensuring the right conditions and incentives are in place to measure peak human capability. This benchmark indicates that existing AI systems still underperform humans. The results of this benchmark were published at the AAAI 2022 conference [124].

Chapter 3 : Neural Type Systems In this chapter, we describe in further detail the problem of Entity Linking within Neuro-Symbolic systems, and present our two-stage approach : 1) discover a type system that associates types to each entity to maximize the task objective, 2) use type prediction as a proxy for the original task objective.

We enable AI systems to represent entities using different levels of abstractions stored in a type system powered by an external knowledge base. We propose to either manually design type systems that provide supervision for a neural network type classifier, or learn how to design the type system that abstracts entities using a mixed integer reformulation of the problem. Each entity is categorized by a set of types (human, country, nationality, etc.).

A neural network is trained to predict for each token in a document the types of the expected entities. The neural network type probabilities serve as soft constraints when deciding among candidate entities.

We propose to measure the discrimination power of a type system by measuring Oracle accuracy : assuming a type classifier perfectly predicts the type constraints, what percentage of entities would be correctly predicted by eliminating those that don't match the expected types ?

Finally, we present experimental results where a system is trained on a large multi-lingual corpora and compared to the state of the art. The proposed system outperforms by a large margin all prior approaches, and identifies using an Oracle a potential performance upper bound that could be reached with better type prediction accuracy. DeepType, the system presented in this chapter was published in at the AAAI 2018 conference [126].

Chapter 4 : End-to-End Type Reasoning In this chapter, we present our approach to end-to-end learning of a Neuro-Symbolic Entity Linking system, and we describe the model's architecture and the details of the experiment.

We further improve the AI system by aligning the training objective with the target task : replace type classification followed by soft constraints by entity classification. To effect this change our neural network now receives as input a variable number of types for each candidate entity and associates a fixed-length vector that can be used during classification.

The types associated with each entity are replaced by Wikidata relation subgraphs. This eliminates the membership rules that were either manually designed or evolved by genetic algorithms. A further benefit of this representational change is the ability to provide entities with autoregressive relational features we call *type interactions*. These features enrich types with higher order information over the full set of entities in a document (e.g. shared employers, geographical co-occurence, list type homogeneity), improving the ability to produce coherent document-wide predictions.

Finally, we summarise our results by comparing the proposed approach to several ablations and the current state of the art, and discuss limitations such as the reliance on structured knowledge bases. The proposed approach is the first to reach super human accuracy at Entity Linking, and outperforms all prior approaches. DeepType 2, the system presented in this chapter was published at the AAAI 2022 conference [124].

Chapter 5 : Neural Relational Database In this chapter, we present our approach for end-to-end learning of a Neuro-Symbolic Entity Linking that can learn from structured and unstructured knowledge bases.

Through a technique we call the Neural Relational Database (NeRD) we extend the available external knowledge available to AI systems to include unstructured relations. This technique works by creating a unified representation for structured and unstructured relations : all relations are first converted to a unified text representation, and second processed by a Pretrained Language Model.

The AI now accesses external knowledge to represent entities using Wikidata relation subgraphs, and also queries relational databases to obtain autoregressive relational features sourced from structured and unstructured relations.

Finally, we present experimental results comparing the proposed approach to several ablations and the current state of the art on Entity Linking. We also provide an approach for using this technique to enhance the capabilities of relation extraction systems.

Outline Discussion

In this chapter we provided an overview of the main areas of progress and present challenges faced by Artificial Intelligence today in making progress towards human-level performance at Natural Language Understanding combined with fine-grained knowledge. We gave a short introduction to the research area of Neuro-Symbolic Artificial Intelligence that seeks to combine the benefits of Neural methods with the advantages of Symbolic Artificial Intelligence to provide new solutions for advancing the field of Artificial Intelligence. Next, we presented the outline of the thesis. This thesis begins with background and context on the Neuro-Symbolic Artificial Intelligence, and each chapter thereafter introduces a different contribution to advance towards our goal of human-level performance Artificial Intelligence by proposing solutions to different challenges in Neuro-Symbolic methods.

In the following chapter we will first describe in greater detail the relevant context for the Neuro-Symbolic work, and second present the state of the art in Artificial Intelligence for the Natural Language Understanding tasks studied.

CHAPTER I

Background and State-of-the-Art

1 . Formal Background

Neuro-Symbolic Artificial Intelligence has become a central focus in the research community, as AI systems become robust enough to interact with the outside world directly. This work aims to further advance Neuro-Symbolic systems to make progress towards AI with human level performance.

This first chapter presents a formal background of the Neuro-Symbolic Artificial Intelligence research area to contextualize for the reader the contributions presented in this thesis. We begin in Subsection 1.1 with the context and motivation, next we present the Neuro-Symbolic Artificial Intelligence area in Section 2. In Section 3 we provide an overview of Natural Language Processing tasks present in this work : Language Modeling in Subsection 3.1, Part of Speech Tagging in Subsection 3.2, Named Entity Recognition in Subsection 3.3, Entity Linking in Subsection 3.4, and Relation Extraction in Subsection 3.5.

1.1 . Context and Motivation

1.1.1 Context

Language emerged in humans some 100,000 to 200,000 years ago and has since evolved into the densest information mediums available [15, 22, 113]. Artificial Intelligence has since its inception had the goal to enable machines to decode human language. Progress has been achieved through a mix of statistical methods and linguistic theory. In the last decade we've witnessed the arrival of machine learning methods that enable machines to match or outperform the accuracy of humans on many linguistic tasks.

The work presented in this thesis continues in this lineage, and presents a methodology for automatically integrating symbolic information into the reasoning process of neural networks that improves accuracy on downstream tasks.

1.1.2 Motivation

Overview This thesis is motivated by finding a solution to an open problem : how to teach Artificial Intelligence systems to interface with external knowledge ? Further motivation for this work comes from identifying a performance gap between humans and machines in the natural language understanding task of Entity Linking. Reaching human-level performance at Entity Linking has proved elusive and especially valuable thanks to its widespread use across intelligent assistants, search engines, translation systems, or automated trading systems.

Closed World Assumption To understand the nature and role of this problem, we must first distinguish between Artificial Intelligence systems that operate with a "Closed World Assumption" (CWA) [131] from those that consider an *open world* [35]. In the context of Machine Learning, the former assumes that the data distribution does not vary. However phenomenons such as concept drift [143] will violate the CWA and can lead the system astray. Techniques that enable intervention or fallbacks have been proposed to mitigate the dangers of drift on CWA models such as online learning, periodic retraining, or Out Of Distribution Detection [10, 178].

Human Level. Despite this strong assumption, this class of Artificial Intelligence systems have become pervasive in our daily lives and across many industries. Adoption of Deep Learning-based approaches was driven by the ability to automate feature learning and construct end-to-end systems that would outperform engineered and heuristically based predecessors across diverse tasks ranging from Speech Recognition [3], Machine Translation [173], Speech Synthesis [48, 7, 164, 116], Image Recognition [59], Image Segmentation [129], or Protein Folding [73].

Computing Power and Gradient Descent Across these various areas, success has been driven by growth in computing power, training data, and model size. Indeed, growth in computing power enabled costly algorithms such as back-propagation to become practical. The back-propagation algorithm developed by different authors in the 1960s and later popularised in the 1980s [75, 139, 167] enabled the discovery of the weights of a neural network by applying the chain rule in a computationally efficient manner over arbitrary computational graphs. Thanks to this algorithm, and its approximate variant Stochastic Gradient Descent [133] which has become the standard for Deep Learning training, it has become possible to learn useful spatial or temporal patterns and features by minimizing an objective function relative to the weights of the neural network with minimal amounts of feature engineering.

Action from Raw Inputs. As the perception capabilities of these learnt systems reach human or superhuman levels, they can now control systems directly from raw inputs with supervision from scalar reward signals. Noteworthy examples of these end-to-end systems include Dexterous Robotic Manipulation [2] trained in simulation and generalizing to real world robots, Game Playing at professional and superhuman levels [109, 160, 147], or Computer Circuit Designs found via reinforcement learning agents that are now being mass-produced [137, 102]. These systems push the limits of the CWA by generalizing from simulation to the reality. They are able to make this jump thanks to high fidelity simulators and techniques such as Domain Randomization [155] that create a training data distribution that is artificially broadened to handle the reality gap.

Open World Assumption A second class of Artificial Intelligence systems are explicitly designed for the *open world*.

Action in a Changing World. Neuro-Symbolic systems are designed with an explicit *open world* assumption by having access to external knowledge [47]. This bridge enables facts, examples, or instructions to be updated live. This capability is particularly desirable as the cost to train leading machine learning models has risen exponentially [4], and their societal impact has increased scrutiny over their reliability [40].

While this direction is promising, it is complicated by the presence of two key operations that are usually discrete and non-differentiable : 1) query design and retrieval, 2) how to process, featurize, and present the received information. Present Neuro-Symbolic systems will therefore not learn these operations but instead rely on manual effort to control queries, retrieval, and integration of results in the model.

Learning to Question.

> If I have seen further it is by standing on the shoulders of Giants.
>
> *Isaac Newton*

The core motivation for this thesis is therefore to support a crucial next step for artificial intelligence systems : be able to ask questions and seek additional information as they reason. This capability is a nod to Isaac Newton, and a parallel to the way humans find answers by reading books, searching the web, or asking others. The volume of information that can be queried is too large for brute force to be practical.

2 . Neuro-Symbolic Artificial Intelligence

2.1 . Definition

Neuro-Symbolic Artificial Intelligence is a sub-area of Artificial Intelligence research focused on finding a synthesis between symbolic Artificial Intelligence and present neural network approaches. A common synthesis involves teaching neural networks to manipulate symbols. Symbol manipulation is a fairly broad concept that can be restricted in the context of this thesis to interacting with databases, graphs, and sets. Interaction with databases includes performing queries, insertions, edits, and deletes. In the case of graphs, a model can check for connectivity between nodes or construct graphs by specifiying edges and nodes. Finally, manipulation and queries of sets enables a model to create set unions or exclusions, and check for set membership.

Making neural predictions symbol aware. Several approaches exist for incorporating symbolic structures into the reasoning process of a neural network. One approach consists in modifying a classification loss function so that it becomes structure aware. In the work of [31] a label hierarchy is used to force a model to make tradeoffs between specificity and accuracy. The objective leverages the hyper/hyponymy relation to make a model aware of different granularity levels. More recently the work of [171] use a hierarchical loss to increase the penalty for distant branches of a label hierarchy using the ultrametric tree distance.

In this thesis, we use different objective functions to incorporate symbolic structures into the reasoning of a neural network. Our approach differs from the work presented above because our we use standard classification losses, but design our labels using the symbolic structure. We also aim to capture the most important aspects of the symbolic structure, however our loss shaping is a result of discrete optimization and incorporates a Learnability heuristic to choose aspects that can easily be acquired.

A different direction for integrating structure stems from constraining model outputs, or enforcing a grammar. In the work of [89], the authors use Named Entity Recognition and FIGER types to ensure that an Entity Linking model follows the constraints given by types. We also use a type system and constrain our model's output, however our type system is task-specific and designed by a machine with a disambiguation accuracy objective, and unlike the authors we find that types improve accuracy. The work of [81] uses a type-aware grammar to constrain the decoding of a neural semantic parser. Our work makes use of type constraints during decoding, however the grammar and types in their system require human engineering to fit each individual semantic parsing task, while our type systems are based on online encyclopedias and ontologies, with applications beyond Entity Linking.

Differentiable Datastructures Starting with recurrent neural networks [71], several forms of learnt memory mechanisms have been proposed to extend a model's context. Noteworthy examples include the fixed-sized memory of the Long Short Term Memory (LSTM) [63] and the addressable-memory of the Neural Turing Machine [52]. Core to these efforts is a differentiable storage and retrieval mechanism. Memory control is learnt using gradient descent end-to-end alongside other model parameters from the same supervision signal. However, the differentiability constraint on insertion, deletion, and retrieval operations leads to some limitations. Fixed-size memories tend to suffer from vanishing gradients or forgetting information when the context window is too large. Variable-length memories struggle with overly large addressable spaces due to over-averaging "memories".

There have been several attempts to extend the memory of a neural network to other data-structures. These include Stacks [72], or Neural Turing Machines [52].

Core to these efforts is a differentiable storage and retrieval mechanism. Scaling the memory to an entire database remains impractical due to the memory and computation requirements.

2.3 . Extending neural network memory to external symbols

Retrieval Models By replacing differentiable retrieval with a nearest neighbor function is then possible to include larger memories. We note the use of episodic memory [132, 44] as a way of rapidly associating states and observations to outcomes and actions. Nearest neighbor retrieval has also been used to broaden the context window of language models in RETRO [14] and ATLAS [68].

In this thesis we attempt to solve a similar problem as episodic memory by enabling neural networks to interface with external databases. This presents the additional challenge of non-differentiable storage and retrieval. Prior work on interfacing with a read–eval–print loop (REPL) solves this challenge by automatically executing queries [37]. Our approach is similar and systematically uses the entities predicted in a document to query a database.

Retrieval Result Representation The next challenge after retrieval is the representation of the query results. Prior approaches include Cognitive Databases, where pretrained word embeddings are used to embed facts [13], but the representation is not task-specific. DrugDBEmbed [9] uses a task-aware Bi-LSTM that produces a column representation to predict drug-drug interactions, but does not allow extending the database.

Our approach to connecting a neural network to a knowledge base is different because it learns a task-specific projection from a task-agnostic Pretrained Language Model representation of the stored facts. Furthermore, the knowledge base can be extended without retraining the neural network : new facts can be inserted into the database by using the Pretrained Language Model's activations followed by the learnt projection.

3 . Natural Language Processing

3.1 . Language Modeling

3.1.1 Definition

Language Modeling is an ambiguous term in Natural Language Processing, representing tasks varying from representation learning to generation. The most commonly accepted definition is as follows :

Definition 3.1: Language Model

A *language model* is a probability distribution over a sequence of words [111]. Progress on this task is measured by either computing the negative log likelihood or perplexity of the target content under the proposed model.

To complete the definition of a language model we also define tokens, the atoms that compose the modeled sequence of words :

Definition 3.2: Token

A *token* is the name given to the atomic components used in a language model. These can range from extremely granular, such as Bytes or characters, to very coarse such as words or phrases.

3.1.2 Background

Origin Starting with Claude Shannon in 1948, the computational task of language modeling begins by focusing on capturing the corpus statistics for words following a sequence of n previous words (n-grams) [145].

Since then Language models have been integral parts of many applications from spelling correction, optical character recognition, to speech recognition [3], and have received heightened attention with the discovery that they can be repurposed for a variety of downstream tasks [19, 23, 119].

Two important directions in language modeling are relevant to this thesis : 1) the use of language models to obtain distributed representations of words and documents, 2) the use of language modeling as a form of pre-training to enable a neural network to generalize better on a downstream task.

Distributed Representations The goal for this subarea of language modeling is to learn semantically rich vectors associated with words, phrases, or documents that support operations useful for downstream tasks. The first example of this technique can be traced to Hinton in [61], however word vectors in natural language processing gained prominence with the publication of the landmark paper *Natural language processing (almost) from scratch* [25] where the authors propose to replace many of the components in the natural language processing toolbox with learnt distributed counterparts. Following this interest in distributed representations, several word vector learning techniques gained popularity such as word2vec [99] and GloVe [112]. Research has since focused on distributed representations produced over sequences using recurrent neural networks such as CoVe [97] and ELMo [115], or Transformers such as BERT [32]. From these works we retain that both supervised and unsupervised tasks are able to train distributed representations of words, phrases, and documents which capture rich semantics. While the quality of the representation varies, it remains remarkable that no ex-

plicit supervision is needed to induce useful topology on the vector spaces such as encoding synonyms and translations nearby in Euclidean distance.

Language Pretraining The use of sequence models such as recurrent neural networks [115, 65] or Transformers [157] for language modeling has created new transfer possibilities for natural language processing tasks in two important waves.

In the first wave, Language models are recast as conditional probability distributions that can be finetuned to produce novel structured outputs outside of the original language modeling task. This finetuning paradigm has powered advances in many downsteam tasks such as translation, parsing, summarization, or sentiment analysis [119, 127, 159, 173].

The second wave identified existing few and zero-shot capabilities in language models without needing to retrain them for a downstream task. Similar to the pre-existing semantic richness found in distributed representations of words and documents, language models have been found to possess an ability to conduct natural language processing tasks by providing examples of the task as context, or describing the task via natural language prompts [19, 119]. This insight has since unlocked the ability to adapt these models using limited data onto new domains, or take advantage of their ability to reference the prompt to provide additional facts or context for "open-book" question answering [134].

3.2 . Part of Speech Tagging

3.2.1 Definition

Part of Speech Tagging or grammatical tagging is the task of assigning to each word in a sentence a specific part of speech (Noun, verb, proposition, etc.). This task has been inextricably linked to Natural Language Processing for many years due to the related difficulties in part of speech disambiguation and meaning. The Brown Corpus [96] has long been the gold standard for evaluation of systems, and has since been supplanted by larger datasets including multilingual ones such as Universal Dependencies [107].

3.2.2 Background

This task has advanced quite a lot from its use of specialized features, to distributed rep, to stack-based. A recent paper [95] comments on the apparent performance ceiling of 97%, presumably tied to some issues with the training data and inconsistencies in the evaluation datasets.

Part of speech tags, are in fact quite broadly useful for downstream applications. For instance, these tags can help detect nominal phrases, or help eliminate spurious predictions in cases where sentences are found to ungrammatical or unsuited for the downstream task.

3.3 . Named Entity Recognition

3.3.1 Definition

This task consists in classifying the nominal groups in a sentence into different coarse-grained categories. The category must be chosen according to the type of entity referenced by the nominal group, such as “person", “organization", “place", etc. Domain specific named entity recognition systems use specialized categories such as “disease" in a medical setting.

3.3.2 Background

Similar to part of speech tagging, Named Entity Recognition is sometimes used as an input to a downstream system, and is considered challenging due to the ambiguities that require understanding of the context to resolve. Within the context of this thesis, Named Entity Recognition tags were used as inputs to an entity linking system [89] and resemble the fine grained types we use in Chapter III.

3.4 . Entity Linking

3.4.1 Definition

Entity Linking has certain key technical terms to describe specific important objects which we will define below. In this task we want to recover the ground truth entities in a knowledge base referred to in a document by specific spans of text called mentions :

Definition 3.3: Mention

A *mention* is a text span in the context of an Entity Linking task that refers to a specific entity within a knowledge base. A document can have multiple mentions, each referring to the same or different entities.

For each mention we want to disambiguate we have to locate the correct referent entity. Commonly, we have a lookup table that maps each mention to a proposal set of n entities for each mention m : $\mathcal{E}_m = \{e_1, \ldots, e_n\}$ (e.g. “Washington" could mean **Washington, D.C.** or **George Washington**). This lookup table is called an alias table :

Definition 3.4: Alias Table

An *alias table* is a lookup table that stores all the possible entities that can be associated to a specific phrase. The Alias Table is typically obtained by using a labeled dataset containing mentions and their entities, and aggregating all the entities shared by the same mention. The term “alias" refers to the fact that the table stores one or more associated entities for each phrase, therefore the phrase is aliased by each of the potential referent entities.

Disambiguation is finding for each mention m the a ground truth entity e^{GT} in $\mathcal{E}_m$. Typically, disambiguation operates according to two criteria : in a large corpus,

how often does a mention point to an entity, $\mathrm{LinkCount}(m, e)$, and how often does entity e_1 co-occur with entity e_2, a proxy for the quality of the predictions which captures the overall *coherence*, but with quadratic computational complexity relative to the number of entities under consideration [101, 41, 175].

3.4.2 Background

The state of the art in entity identification and disambiguation can be structured along several dimensions we discuss below.

Abstract Entity Representations and Types. The work of [89] uses the diverse types of Named Entity Recognition tags (e.g., persons, places) to categorize all candidate entities in their Entity Linking system. The use of abstract entities was further generalized in DeepType [126], considering all Wikidata classes as potential categories, or types, and shows a type predictor suffices to disambiguate. Abstract description-based representations are also used in [93, 106]. In [104], the proposed Entity Linking system combines pre-trained language models with entities described by a transcription of their Wikidata relations.

Identification and Disambiguation Loss. Most approaches rely on either generative or constrastive losses. In the former case, the sought model is optimized to maximize the log-likelihood of the ground truth interpretation. In the latter case, the model is optimized to enforce a sufficient margin between the ground truth interpretation and alternatives [55]. The two approaches have complementary strengths and weaknesses. The generative approach is based on first principles ; it enables to assess any interpretation at the expense of a (very) high sample complexity ; the challenge is to define the search space. The contrastive approach, only aims at making the good interpretation the preferred one by only requiring that the different input spaces (images, text, knowledge graph nodes) project into a mutual scalar comparison space [106].

SoTA and Attention. A recent trend in Entity Linking systems is instead to perform independent predictions but use a pretrained language models with attention to ensure long-range context informs each prediction [106, 172, 104, 87]. The features from language modeling help to ensure the model learns a rich textual encoding, and also reduces the chances of overfitting when transferring a model from a high supervision regime (language modeling) to a sparsely supervised setting (Entity Linking). The high memory and computation cost limit the applicability of these models to long documents. The current SoTA [42] circumvents this issue by truncating the document to keep a window around a mention. This approach approximates global context by gluing back the document title to the window.

Coherency and Relational Information. Entity Linking selects entities based on their individual relevance, where a key component is their compatibility with the other document entities. In multi-mention documents, jointly predicting entities can be helpful to improve coherency or use information from other predictions to assist later disambiguation decisions [169, 101]. The connections between entities can be measured using reciprocal link statistics from Wikipedia [101], by analyzing the link graph using a PageRank algorithm [114, 53]. Another direction aims to learn distributed entity representations using random walks or by using negative sampling that also capture a measure of coherence [175, 49, 176, 84].

As the number of potential entity pairs is large, computing coherence metrics presents a computational challenge. In [49] the authors use attention over a subset of the document mentions to reduce the computational cost. Most similar to the work in this thesis is the textual transcription of an entity's structured relations in [104]. This approach also uses relational information to represent entities but only makes use of structured relations. Our approach uses relational information between entities as well as attention over the mentions to improve coherence. Unlike prior approaches, in DeepType 3 we also exploit unstructured relations to further improve performance.

3.5 . Relation Extraction

3.5.1 Definition

The goal in this task is to predict the presence or absence of a typed directed *relation* between pairs of highlighted phrases (*mentions*) in a document. The possible relation types change depending on the target domain or dataset. In this thesis we focus our experiments on the DocRED [180] dataset where the relation types are chosen among 96 Wikidata relations.

3.5.2 Background

Open information extraction [39] aims to collect unstructured relations from natural language corpora. A triplet representation ("subject text", "relation text", "object text") makes it way into an Open Knowledge Base (KB) [45]. The unstructured relations can then be mapped to structured relations to perform link prediction [18]. The unstructured relations stored in NeRD resemble those in the Open KB, but downstream usage by NeRD operates on them directly without requiring any alignment or mapping back to structured relations.

Relation Extraction using Relational Information In Relation Extraction [24, 183] the goal is detect structured relations between pairs of phrases (mentions) in natural language text. Information about mention entities has previously been shown to improve relation extraction performance [36], or helpful when jointly predicting entities and relations [86]. The state of the art [174] in this task uses a pretrained language model as its text representation, but does augment its mention representation with entity information. Our work builds upon both of these ideas by refining the relation predictions from [182, 183] using entity information.

CHAPTER II

Human Performance Benchmark

1 . Introduction

Benchmarks play a crucial role in measuring progress in Artificial Intelligence (AI) towards human-level performance. The earliest AI benchmarks were designed with the idea of a live presence of humans for evaluation, such as the “Turing test" also known as the “Imitation Game" [156], or competitive games such as Chess with Deep Blue [66] playing against Garry Kasparov [166], in Go with AlphaGo [147] playing against Lee Sedol [142], or in Dota 2 with OpenAI Five [109] playing against OG [152].

With the rise of Machine Learning, a desire for repeatable experimentation and fair methodologies to compare the work of AI researchers motivated the creation of standard datasets and benchmarks. Certain benchmarks have had an outsized impact on the AI and Machine Learning fields such as the object classification dataset Imagenet [30].

Within Natural Language processing, benchmarks serve to highlight shortcomings in existing systems, and also to bring attention to specific areas by creating opportunities for publication and collaboration. The National Institute of Standards and Technology's Text Analysis Conference (TAC) challenges started in 2008 [28] has included tracks to focus on specific topic areas such as the Knowledge-Base Population track (TAC-KBP) used extensively in this thesis. Each year, the tracks are updated to reflect progress or improvements to the framing. More recently, a series of challenges have been designed to elucidate unique common-sense reasoning problems and tasks such as LAMBADA [110] which was a notoriously challenging question answering task until the arrival of large-scale pretrained in the form of GPT [120], GPT-2 [121], and GPT-3 [19]. As a consequence of the fall of previous benchmarks, new ones were proposed such as GLUE [161] and SuperGLUE [162]. The benchmark we propose in this chapter is a continuation of this lineage, where we identify a need for a new measurement because of saturation of earlier benchmarks and to bring focus to aspects of natural language understanding that are still challenging for machines.

Entity Linking is an area where human performance has not been established or benchmarked, and where AI system performance has seemingly plateaued. Researchers in the Entity Linking field have extended evaluations using new Entity Linking datasets to focus on rare entities such as WikilinksNED [38], new contexts such as dialog with CREL [70], or multilingual entity linking such as VoxEL [135], but do not reveal information on whether AI systems are already at the limit of human performance or beyond. This gap motivates the creation of the first human performance benchmark for Entity Linking, a task that requires a combination of fine grained natural language understanding and knowledge of real-world facts.

In this chapter we present our human performance benchmark for Entity Linking by studying the accuracy of a panel of humans on two standard and widely studied datasets TAC KBP 2010 [69] (TAC) and CoNLL AIDA (YAGO) [64]. The first section presents the approach, including the selection, annotation, and technique

used to combine responses ; the second section presents the results and observations from the benchmark, and third we conclude with a discussion of the results and perspectives for future work and how this fits within the rest of the thesis.

2 . Approach

In this section we present our approach to design a human performance benchmark for Entity Linking. First, we describe the steps involved in preparing the data for the benchmark and human consumption. Second, we explain how we constructed a panel of human annotators. Third, we introduce the software and interface used to collect the results. Fourth, we introduce our methodology for measuring human accuracy and agreement.

2.1 . Data preparation

Word	Boundary	Wikipedia URL
EU	B	
rejects		
German	B	/Germany
call		
to		
boycott		
British	B	/United_Kingdom
lamb		
.		
Peter	B	
Blackburn	I	
BRUSSELS	B	/Brussels

FIGURE II.1 – Sample data from CoNLL-AIDA (YAGO) dataset containing documents structured with the location of mentions and the associated entity using a Wikipedia URL. The boundary column is used to indicate starts ("B") and continuations ("I") of mentions that must be linked to an entity, while the Wikipedia URL column provides a reference to the correct entity in the Wikipedia knowledge base.

We design our human performance benchmark for Entity Linking using widely studied datasets to facilitate comparisons with prior work and maximize the impact on future research by ensuring the results are obtained on accessible and trusted datasets. While the datasets were obtained by asking human annotators to decide what entity specific phrases (mentions) in a document corresponded to, the output of this annotation was designed for machine consumption, not to create a human readable multi-choice questionnaire. In an example of the CoNLL-AIDA dataset

shown in Figure II.1 we can immediate notice how the entity selection will be challenging for a human annotator : is the selection of entities be done by writing manually the Wikipedia URL, by selecting from a list of all possible Wikipedia pages, or perhaps by providing the correct answer along with a few distractor options ?

In our work we've chosen to help human annotators by making the list of entities to choose from be selected using an alias table, a mapping we build that tells us for each mentions what potential entities that have previously been associated. Thanks to this approach, the number of potential options drops from 40M to less than 10 on average. We further assist annotators by taking into account the popularity and number of inbound links that each entity has on Wikipedia to sort the options with the most common entity shown first. These choices make the selection process easier for a human, but mostly they ensure that each decision can be taken in a reasonable amount of time by making the task closer to asking : "is the default meaning correct, or do you want to look for a slightly less common meaning for the mention ?" We experimented with turning off the ordering and found that human performance dropped precipitously, thereby suggesting that there is a strong priming effect given by ordering according to popularity.

A final consideration for converting the Entity Linking task from a machine to a human-friendly task is whether annotators must disambiguate all the entities in a document one by one or all at once. The argument in favor of having annotation be done all at once comes from AI systems that perform better they jointly disambiguate entities and increase coherency. However, joint annotation is impractical for human annotators for a variety of reasons. First, if we want to use crowdsourcing platforms such as Amazon Mechanical Turk to perform the annotation, then breaking the task into small chunks that require a similar amount of time and effort is a crucial for the work to be done in a timely and reliable manner. Second, documents in the datasets vary widely in length and number of mentions that have to be disambiguated, therefore it will be hard to isolate whether the accuracy of the results will be biased positively or negatively by the length of the document or the inherent difficult of the specific document. Indeded, certain documents contain multiple mentions : in AIDA there are on average 15 per document. Because of this reasons, we choose to standardize the task by having workers disambiguate each mention in a document independently. Fortunately, we can amortize document reading time by offering all tasks from the same document to the annotator in consecutive order.

2.2 . Human Panel Selection

A key component of human annotated benchmarks and datasets is the reliability and quality of the labels. A particularly effective way to obtain high quality labels on larger datasets is to use crowdsourcing services such as Amazon Mechanical Turk or Crowflower. The quality of the responses varies of course with the expertise of the annotator, the amount of time given to respond, and the clarity of the task. In order for our benchmark to properly measure human performance we need to ensure that workers are selected to be knowledgeable, understand the goal of the annotation, and are sufficiently well incentivized to complete the task correctly and with care.

2.2.1 Screening

We first improve response quality by taking particular care to screen, and brief : they must be native English speakers and have Amazon Mechanical Turk's Master qualification, a recognition of prior excellence in annotation tasks.

A common strategy to further improve quality is to select participants using a trial stage and keep only the top performing ones for the actual benchmark. A second technique that is used for crowdsourcing quality improvement is to include a test question and answer that verifies whether the annotators read the instructions. In our work we apply both of these strategies : we took 10 documents and asked all participants meeting the language and Master qualification to respond, and kept those that did best on this trial portion.

2.2.2 Incentives

We are able to further improve annotation quality through a special incentive structure we can put in place. Commonly crowdsourcing platforms are used to label new data where it is hard to know with certainty whether the annotators are making mistakes. Luckily, because we are asking annotators to relabel existing datasets, the correct answer is already known therefore we can immediately detect when an annotator is performing well. Thanks to this hidden label information, we can further increase the incentive to provide accurate answers thanks to a bonus that is paid out only when an annotator gets the right answer.

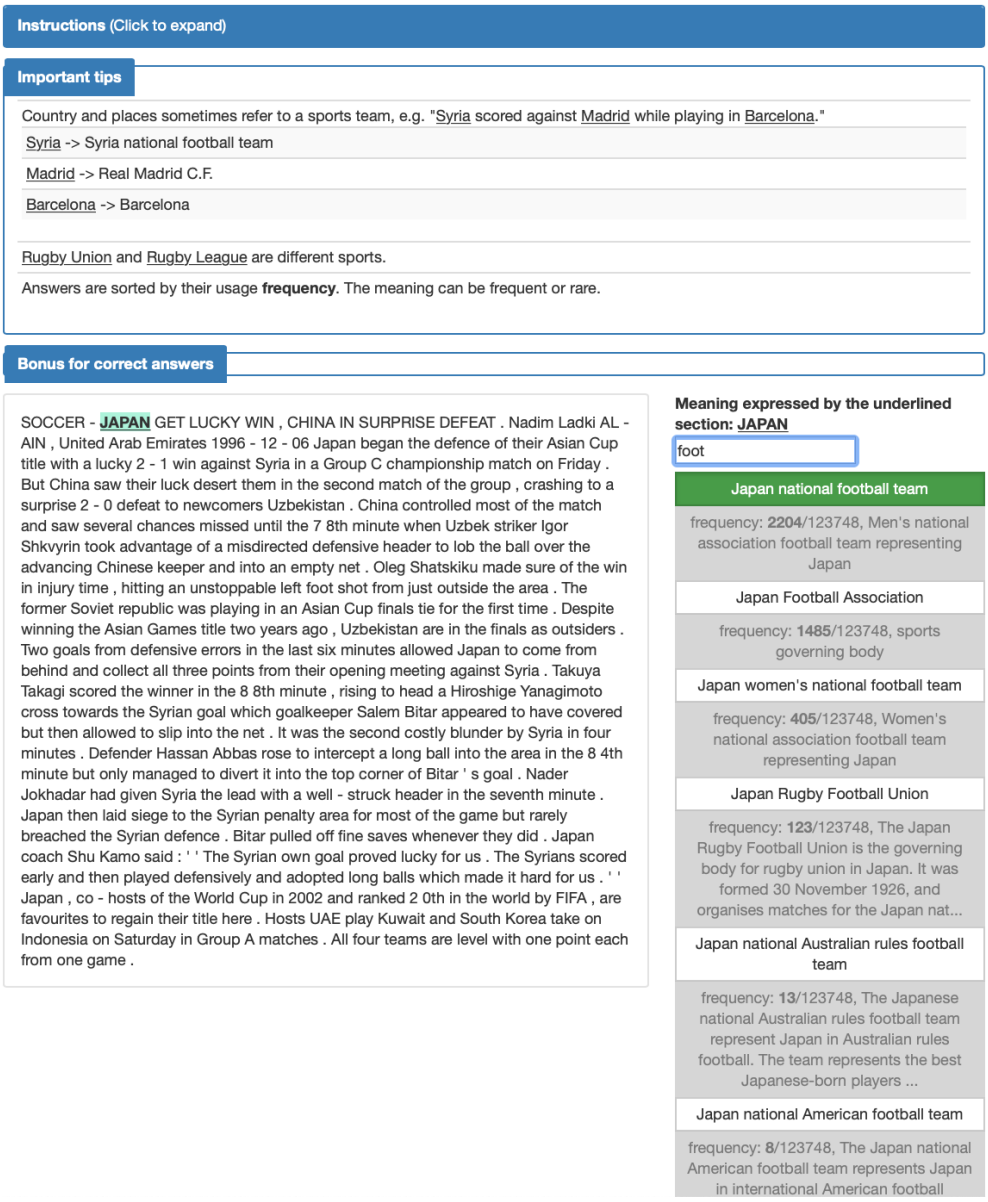


FIGURE II.2 – The annotation interface in Amazon Mechanical Turk shows a single highlighted mention at a time. Options are shown in a list with descriptions, title, and link frequency stats. To assist annotators the results are ordered by link frequency and a full-text search bar enables quick filtering of the options. Instructions and tips are shown at the top of the page. We check whether annotators click to expand the instructions to find the most thorough annotators.

2.3 . Annotation Interface

We develop a webpage interface to receive annotations in the Amazon Mechanical Turk platform. In this interface, each labelled mention is presented to humans by highlighting the mention within the full original document. Annotators select candidate entities from a list generated given by a Wikipedia alias table [1]. To assist the annotator, candidate entities are shown with their full title, Wikipedia description and usage frequency, and ordered by their Wikipedia link frequency as visible in the screenshot shown in Figure II.2.

2.4 . Annotator vs. Panel Accuracy

A potential source of error in our benchmark may come from differences in annotator expertise, factual knowledge, or effort. Despite the efforts described earlier to screen and incentivize annotators, annotators may still have inconsistencies in their familarity with different subject matters found in Entity Linking datasets varying from sports, to politics, to technology. To reduce the effect of annotator expertise differences, we assign each labelled mention to three different annotators and measure accuracy by checking whether any of the three annotators got the answer correct. Because this measurement requires us to act like an oracle that knows the correct answer, we name this metric the "Human Oracle accuracy".

3 . Results

3.1 . Human vs. AI Accuracy

Model	TAC	AIDA
Human Oracle accuracy	**96.86**	**96.78**
Human Majority accuracy	95.39	93.35
Ling et al. [87]	89.8	94.9
Mulang' et al. [104]	-	94.94
Févry et al. [42]	94.9	96.7

TABLE II.1 – Humans and state of the art EL system accuracy (best results in bold).

We conduct the human performance benchmark and find that humans (Human Oracle Accuracy) reach 96.86% on TAC and 96.78% on AIDA, outperforming the current state of the art on these tasks as shown in Table II.1. We observe an accuracy gap remains between a human panel and prior algorithmic approaches of 1.96% on TAC and 0.08% on AIDA leaving room for algorithmic improvement. Humans have similar accuracy on TAC and AIDA, while surprisingly SoTA algorithmic approaches until 2020 perform 4.09% higher on AIDA than TAC.

1. The alias table always contains the correct answer.

We then study whether the benchmark's annotators produce consistent results by measuring their agreement. Fleiss's κ [43] is a standard metric used for measuring whether annotators make consistent decisions with each other. Fleiss's κ ranges from 0 to 1, with 1 indicating perfect agreement, and 0 indicating that annotators disagree. We find that agreement is high (0.9396 on AIDA and 0.9684 on TAC), supporting the claim that annotators reached similar conclusions and did not respond randomly. The agreement and mean response time per dataset are given in Table II.2.

	TAC	AIDA
Agreement (Fleiss' κ)	0.9684	0.9396
Mean response time (seconds)	18.49	15.68
Participants	255	5
Total Responses	2,203	13,934

TABLE II.2 – Statistics showing the inter-annotator agreement (Fleiss's κ) and mean response time for both datasets in the benchmarks. We notice that agreement is high (close to 1) indicating that the results collected do not show evidence of spurious or random responses.

3.2 . Response Time

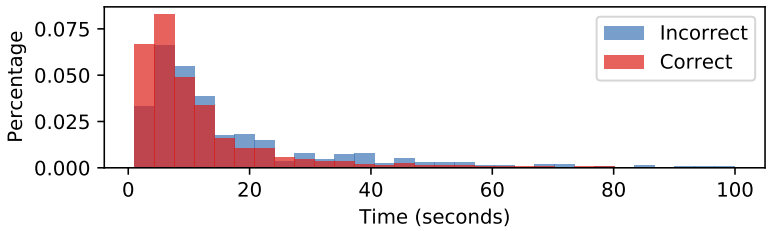


FIGURE II.3 – Time taken by AMT annotators grouped by correct and incorrect response times.

We record the time taken by annotators on each mention and do not detect a significant timing difference between correct and incorrect responses reducing the possibility that mistakes were caused by clicking errors or gaming the task. A histogram of the timings is shown in Figure II.3. Based on the time taken to answer, compensation is 9.73/11.47 $/hour for TAC/AIDA, and with bonuses is 15.57/18.36 $/hour. In total, $1,307 were spent on worker wages.

3.3 . Participant Risk and Review Board

The study presented in this work presents minimal risk to the human participants and does not ask or collect personally identifying information. As such, the study meets the IRB exempt status (US45CFR46 §46.101). Specifically, we collect answers to multiple choice questions along with the elapsed time.

The source material is also neutral in tone and inoffensive. We use news articles covering neutral topics such as sports results, tabloids, and news dispatches from the frequently studied datasets CoNLL AIDA (YAGO) dataset and TAC-KBP 2010.

4 . Discussion

4.1 . Overview

In this chapter we introduced the first human performance benchmark for Entity Linking. Several key difficulties needed to be overcome to construct the benchmark. We first converted a task designed for AI systems so that humans can participate. Second, we took special care through screening, educating, and incentivizing mechanical turk workers to enable crowdsourced responses to faithfully measure peak human performance. Third, a new interface and tool had to be built to support the collection of entity selection responses where the context is complex and rich, and where there are thousands of potential options to select from in each question. Our results show that the task was made sufficiently accessible for participants to answer with high inter-annotator agreement, and obtain accuracies that are higher than present systems.

4.2 . Future Work

Establishing this benchmark opens several directions for future work. First, the results motivate further research into Natural Language Processing on Entity Linking given that we detect a gap between present AI systems and human performance. One of the goals of thesis is to close this gap, which we do by obtaining new state of the art results in Chapter III, and ultimately superhuman performance in Chapter IV and Chapter V. Second, as part of the publication of DeepType 2 [124], we open-sourced the benchmark at this url [2] enabling others to perform in-depth error analysis, or build new benchmarks on other languages or datasets thanks to the framework and software we release.

2. https://github.com/deep-type/deeptype2

4.3 . Relation to Thesis

In the following chapter (Chapter III), we develop DeepType, a first AI system that takes advantage of human knowledge bases and hierarchies of concepts which dramatically simplifies the task of Entity Linking. As we shall see, abstracting entities using “types" that are built from the existing concept hierarchies and knowledge bases enables us to build Entity Linking systems that learn to distinguish entities purely through what “type" they belong to. Representing entities through types has two advantages : 1) the task is simplified since there are many fewer types (thousands) than entities (40 million), 2) thanks to the benchmark in this chapter, we can observe that systems built with this simplified view of entities have a performance upper bound that is slightly higher than human performance, showing that there is sufficient discrimination power with pure “types" to build AI systems that rival humans at Entity Linking performance.

CHAPTER III

Neural Type Systems

1 . Introduction

Online encyclopaedias, knowledge bases, ontologies (e.g. Wikipedia, Wikidata, Wordnet), alongside image and video datasets with their associated label and category hierarchies (e.g. Imagenet [30], Youtube-8M [1], Kinetics [74]) offer an unprecedented opportunity for incorporating symbolic representations within distributed and neural representations in Artificial Intelligence systems. Several approaches exist for integrating rich symbolic structures within the behavior of neural networks : a label hierarchy aware loss function that relies on the ultrametric tree distance between labels (e.g. it is worse to confuse sheepdogs and skyscrapers than it is to confuse sheepdogs and poodles) [171], a loss function that trades off specificity for accuracy by incorporating hypo/hypernymy relations [31], using Named Entity Recognition types to constrain the behavior of an Entity Linking system [89], or more recently integrating explicit type constraints within a decoder's grammar for neural semantic parsing [81]. However, current approaches face several difficulties :

— Selection of the right symbolic information based on the utility or information gain for a target task.

— Design of the representation for symbolic information (hierarchy, grammar, constraints).

— Hand-labelling large amounts of data.

DeepType overcomes these difficulties by explicitly integrating symbolic information into the reasoning process of a neural network with a type system that is automatically designed without human effort for a target task. This is achieved by reformulating the design problem into a mixed integer problem : create a type system by selecting roots and edges from an ontology that serve as types in a type system, and subsequently train a neural network with it. The original problem cannot be solved exactly, so a 2-step algorithm is used instead :

1. heuristic search or stochastic optimization over the discrete variable assignments controlling type system design, using an Oracle and a Learnability heuristic to ensure that design decisions will be easy to learn by a neural network, and will provide improvements on the target task,
2. gradient descent to fit classifier parameters to predict the behavior of the type system.

In order to validate the benefits of our approach, we focus on applying DeepType to Entity Linking, the task of resolving ambiguous mentions of entities to their referent entities in a knowledge base (KB) (e.g. Wikipedia). Specifically we compare our results to prior approaches on three standard datasets (WikiDisamb30, CoNLL (YAGO), TAC KBP 2010). We verify whether our approach can work in multiple languages, and whether optimization of the type system for a particular

language generalizes to other languages [1] by training our full system in a monolingual (English) and bilingual setup (English and French), and also evaluate our Oracle (performance upper bound) on German and Spanish test datasets. We compare stochastic optimization and heuristic search to solve our mixed integer problem by comparing the final performance of systems whose type systems came from different search methodologies. We also investigate whether symbolic information is captured by using DeepType as pretraining for Named Entity Recognition on two standard datasets (i.e. CoNLL 2003 [141], OntoNotes 5.0 (CoNLL 2012) [118]).

Our key contributions in this chapter are as follows :

— A system for integrating symbolic knowledge into the reasoning process of a neural network through a type system, to constrain the behavior to respect the desired symbolic structure, and automatically design the type system without human effort.

— An approach to Entity Linking that uses type constraints, reduces disambiguation resolution complexity for a document with N mentions from $O(N^2)$ to $O(N)$.

— The ability to incorporates new entities into the system without retraining.

— Outperforms all prior solutions to Entity Linking by a wide margin.

Moreover, we observe that disambiguation accuracy reaches 99.0% on CoNLL (YAGO) and 98.6% on TAC KBP 2010 when entity types are predicted by an Oracle, suggesting that Entity Linking would be almost solved if type prediction accuracy can be improved.

The rest of this chapter is structured as follows. In Section 2 we explain how types help us reframe the general Entity Linking task into Entity Linking with Types as soft constraints. In Section 3 we present our approach. In Section 4 we provide experimental results for DeepType applied to Entity Linking and evidence of cross-lingual and cross-domain transfer of the representation learned by a DeepType system. A discussion of results and directions for future work are given in Section 5.

2 . Entity Linking with Soft Constraints

DeepType can be used to constrain the outputs of a neural network using a type system by extending the Entity Linking task to associate with each entity a series of types (e.g. `Person`, `Place`, etc.) that if known, would rule out invalid answers, and therefore ease linking (e.g. the context now enables types to disambiguate "Washington"). Knowledge of the types T associated with a mention can also help prune entities from the candidate set, to produce a constrained set : $\mathcal{E}_{m,T} \subseteq \mathcal{E}_m$. In a probabilistic setting it is also possible to rank an entity e in document x

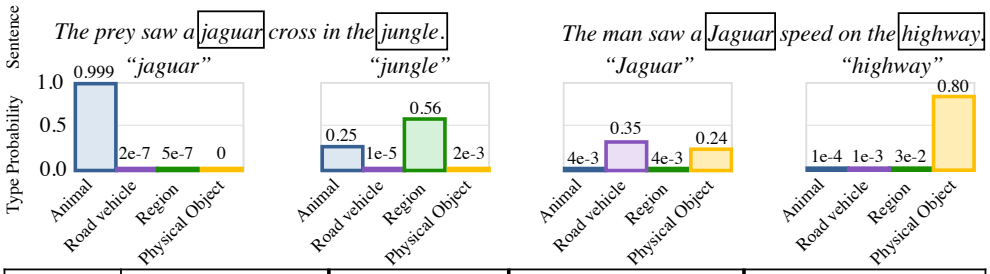
1. e.g. Does it overfit to a particular set of symbolic structures useful only in English, or can it discover a knowledge representation that works across languages ?

according to its likelihood under the type system prediction and under the entity model :

$$\mathbb{P}(e|x) \propto \mathbb{P}_{\text{type}}(\text{types}(e)|x) \cdot \mathbb{P}_{\text{entity}}(e|x, \text{types}(e)). \quad (1)$$

In prior work, the 112 FIGER Types [88] were associated with entities to combine an Named Entity Recognition tagger with an Entity Linking system [89]. In their work, they found that regular Named Entity Recognition types were unhelpful, while finer grain FIGER types improved system performance.

3 . Approach



Entity	jaguar	Jaguar	jungle	jungle	jaguar	Jaguar	highway	Highway
Type	Animal	Road vehicle	Region	Music	Animal	Road vehicle	Physical Object	Film
only link Prob.	0.29	**0.60**	**0.35**	0.17	0.29	**0.60**	**0.85**	0.04
Prob. w/. types	**1.0**	0.0	**1.0**	0.0	0.0	**1.0**	**1.0**	0.0

FIGURE III.1 – Example model output : "jaguar" refers to a different entity (car or animal) depending on the context. Predicting the type associated with each word (e.g. animal, region, etc.) helps eliminate options that do not match, and recover the true entity. Bar charts give the system's belief over the type-axis "`IsA`", and the table shows how types affects entity probabilities given by Wikipedia links. Each column of the table is a different entity, and highest probability is bolded.

DeepType is a technique for integrating symbolic knowledge into the reasoning process of a neural network through a type system. When we apply this technique to Entity Linking, we constrain the behavior of an entity prediction model to respect the symbolic structure defined by types. As an example, when attempting to disambiguate "Jaguar" the benefits of this approach are apparent : the decision can be based on whether the predicted type is Animal or Road Vehicle as shown visually in Figure III.1.

In this section, we will first define key terminology, then explain the model and its sub-components separately.

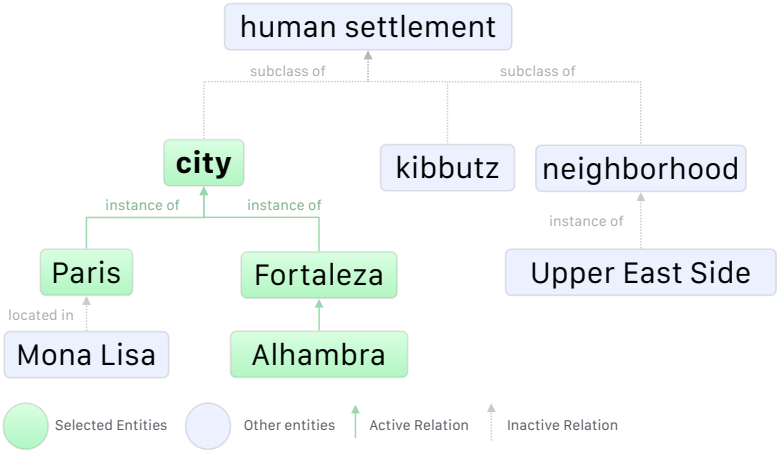


FIGURE III.2 – Defining group membership with a knowledge graph relation : children of root (city) via edge (instance of). In this example we are looking at all children of the entity "city" (in bold) that are connected via the "instance of" relation. The selected children visible in this example in green are Paris, Fortaleza, Alhambra. The entities that were not selected are shown in grey.

Definition 3.1: Relation

Given some knowledge graph or feature set, a *relation* is a set of inheritance rules that define membership or exclusion from a particular group. For instance the relation `instance of(`**city**`)` selects all children of the *root* **city** connected by `instance of` as members of the group, depicted by green boxes in Figure III.2.

Definition 3.2: Type

In this work a *type* is a label defined by a *relation* (e.g. `IsHuman` is the type applied to all children of **Human** connected by `instance of`).

Definition 3.3: Type Axis

A *type axis* is a set of mutually exclusive types. Examples of type axes include geographical categories (North America, South America, Europe, etc.), typological categories (human, animal, vehicle, company, etc.), occupational categories (politician, writer, pilot, doctor, etc.), or even topical categories (science, geography, fiction, medicine, history).

Definition 3.4: Type System

A *type system* is a grouping of *type axes*, $\mathcal{A}$. For instance a type system with two axes {`IsA`, `Topic`} assigns to **George Washington** : {`Person`, `Politics`}, and to **Washington, D.C.** : {`Place`, `Geography`}). In the example shown in Figure III.2 a type system distinguishes the entities that are children of "city" from other entities.

Definition 3.5: Oracle

The *Oracle* is a methodology for abstracting away machine learning performance from the underlying representational power of a type system $\mathcal{A}$. It operates on a test corpus with a set of mentions, and the associated true entities e^{GT}, and the candidate set of entities for the mention : $m_i, e_i^{\text{GT}}, \mathcal{E}_{m_i}$. The Oracle prunes each candidate set to only contain entities whose types match those of e_i^{GT}, yielding $\mathcal{E}_{m,\text{oracle}}$. Types fully disambiguate when $|\mathcal{E}_{m,\text{oracle}}| = 1$, otherwise the entity prediction model is used to select the right entity in the remainder set $\mathcal{E}_{m_i,\text{oracle}}$:

$$\text{Oracle}(m) = \underset{e \in \mathcal{E}_{m,\text{oracle}}}{\text{argmax}} \ \mathbb{P}_{\text{entity}}(e|m, \text{types}(x)). \tag{2}$$

If $\text{Oracle}(m) = e^{\text{GT}}$, the mention is disambiguated. Oracle accuracy is denoted S_{oracle} given a type system over a test corpus containing mentions $M = \{(m_0, e_0^{\text{GT}}, \mathcal{E}_{m_0}), \ldots, (m_n, e_n^{\text{GT}}, \mathcal{E}_{m_n})\}$:

$$S_{\text{oracle}} = \frac{\sum_{(m,e^{\text{GT}},\mathcal{E}_m) \in M} \mathbb{1}_{e^{\text{GT}}}(\text{Oracle}(m))}{|M|}. \tag{3}$$

3.2 . Model

To construct an Entity Linking system that uses type constraints this requires : a type system, the associated type classifier, and a model for predicting and ranking entities given a mention. Instead of assuming a joint type system, classifier, and entity prediction model, we will instead create the type system and its classifier starting from a given entity prediction model and ontology with text snippets containing entity mentions (e.g. Wikidata and Wikipedia). For simplicity we use $\text{LinkCount}(e, m)$ as our entity prediction model.

We restrict the types in our type systems to use a set of parent-child relations over the ontology in Wikipedia and Wikidata, where each type axis has a root node and an edge type, that sets membership or exclusion from the axis. For instance, if we use as root the entity "human" and the edge type "instance of" then entities are split into human and non-human entities [2]).

2. The descendants of "human" via the "instance of" relation are effectively labeled the same way as the Named Entity Recognition label for person (PER).

We then reformulate the problem into a mixed integer problem, where discrete variables control which roots entities $e_1, \ldots, e_k$ and relation edges $r_1, \ldots, r_k$ among all entities and relations will define type axes. A classifier parametrized by θ is fit to the type system by predicting for each appearance of an entity in a corpus, the associated inheritance labels produced by the type system. The goal in type system design is to select parent-child relations that a classifier easily predicts, and where the types improve disambiguation accuracy.

3.3 . Objective

To formally define our mixed integer problem, we first denote $\mathcal{A}$ as the assignment for the discrete variables that define our type system (i.e. boolean variables defining if a parent-child relation gets included in our type system), θ as the neural network weights for our entity prediction model and type classifier, and $S_{\text{model}}(\mathcal{A}, \theta)$ as the disambiguation accuracy given a test corpus containing labeled mentions m with their true entity e^{GT}. We denote the set of mentions and their associated entities in a document as $M = \left\{(m_0, e_0^{\text{GT}}, \mathcal{E}_{m_0}), \ldots, (m_n, e_n^{\text{GT}}, \mathcal{E}_{m_n})\right\}$. We now assume our model produces some score for each proposed entity e given a mention m in a document D, defined $\text{EntityScore}(e, m, D, \mathcal{A}, \theta)$. The predicted entity for a given mention is thus :

$$e^* = \underset{e \in \mathcal{E}_m}{\text{argmax}}\, \text{EntityScore}(e, m, D, \mathcal{A}, \theta). \tag{4}$$

If $e^* = e^{\text{GT}}$, the mention is disambiguated. Our problem is thus defined as the following optimization problem :

$$\max_{\mathcal{A}} \max_{\theta}\ S_{\text{model}}(\mathcal{A}, \theta) = \frac{\sum\limits_{(m, e^{\text{GT}}, \mathcal{E}_m) \in M} \mathbb{1}_{e^{\text{GT}}}(e^*)}{|M|}. \tag{5}$$

This original formulation cannot be solved exactly [3]. To make this problem tractable we propose a 2-step algorithm :

1. **Discrete Optimization of Type System** : Heuristic search or stochastic optimization over the discrete variables of the type system, $\mathcal{A}$, informed by a Learnability heuristic and an Oracle.
2. **Type classifier** : Gradient descent over continuous variables θ to fit type classifier and entity prediction model.

We will now explain in more detail discrete optimization of a type system, our heuristics (Oracle and Learnability heuristic), the type classifier, and prediction in this model.

3. There are nearly $2^{2.4 \cdot 10^7}$ choices if each Wikipedia article can be a type within our type system.

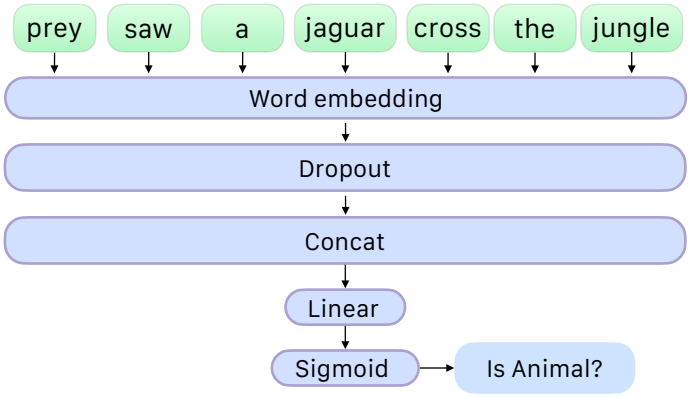


FIGURE III.3 – A neural network that receives a limited window of words around a center-word for binary classification of membership in a particular type. In this example the sentence "prey saw a jaguar cross the jungle" is cut into words, each of which is turned into a distributed representation using word embeddings. The word vectors from the window are concatenated into a single vector that is then fed to a linear classifier whose output indicates membership. In this example we are looking for whether the entity linked by the word "jaguar" would belong to the "Is Animal" type.

Type Systems can vary in their achievable Oracle accuracy $S_{\text{oracle}}(\mathcal{A})$ but also in how easy they translate to a real trainable type classifier. To ensure that the disambiguation gains observed using the measurement $S_{\text{oracle}}(\mathcal{A})$ are actually reachable by a trained type classifier, the selected types must have some kind of guarantee that they can be predicted. The Learnability heuristic empirically measures the average performance of classifiers at predicting the presence of a type within some Learnability-specific training set.

To efficiently estimate Learnability for a full type system we make an independence assumption and model it as the mean of the Learnability for each individual axis, ignoring positive or negative transfer effects between different type axes. This assumption enables parallel training of simpler classifiers for each type axis. We measure the Area Under its receiver operating characteristics Curve (AUC) for each classifier and compute the type system's learnability :

$$\text{Learnability}(\mathcal{A}) = \frac{\sum_{t \in \mathcal{A}} \text{AUC}(t)}{|\mathcal{A}|}. \tag{6}$$

We use a text window classifier trained over windows of 10 words before and after a mention. Words are represented with randomly initialized word embeddings ; the classifier is illustrated in Figure III.3. AUC is averaged over 4 training runs for each type axis.

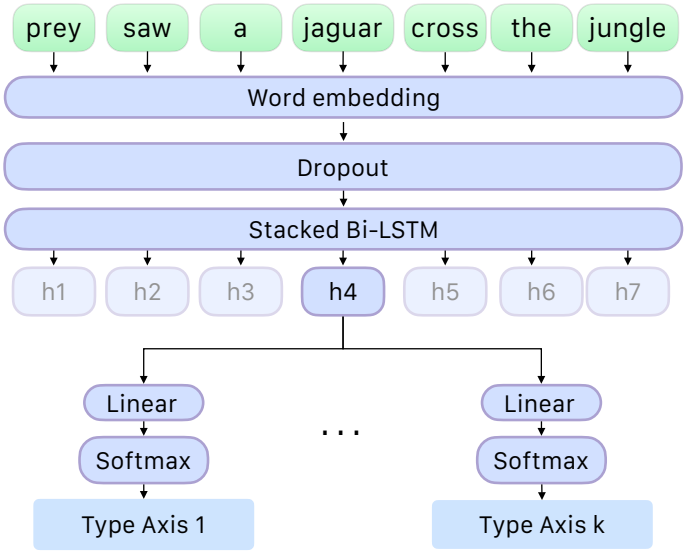


FIGURE III.4 – Neural network architecture for DeepType that discovers long-term dependencies to predict types and jointly produces a distribution for multiple type axes. In this example the sentence "prey saw a jaguar cross the jungle" is cut into individual words. Each word is sent to a word embedding layer produced a distributed representation for each word. The word vectors are processed in sequence by a bidirectional-LSTM neural network. The hidden states produced by the bidirectional-LSTM corresponding to each word in the sequence can then be individually classified using k different classifiers, each corresponding to a different Type Axis. Each Type Axis handles decisions among exclusive options such as selecting : geographical class such as North America vs. Europe, typological class such as human vs. company vs. automobile, etc.

3.5 . Discrete Optimization of a Type System

The original objective $S_{\text{model}}(\mathcal{A}, \theta)$ cannot be solved exactly, thus heuristic search or stochastic optimization are necessary to find suitable assignments for $\mathcal{A}$. To avoid training an entire type classifier and entity prediction model for each evaluation of the objective function, we instead use a proxy objective J for the discrete optimization [4]. To ensure that maximizing $J(\mathcal{A})$ also maximizes $S_{\text{model}}(\mathcal{A}, \theta)$, we introduce a Learnability heuristic and an Oracle that quantify the disambiguation power of a proposed type system, an estimate of how learnable the type axes in the selected solution will be. We measure an upper bound for the disambiguation power by measuring disambiguation accuracy S_{oracle} for a type classifier Oracle

4. Training of the type classifier takes about 3 days on a Titan X Pascal, while our Oracle can run over the test set in 100ms.

over a test corpus.

To ensure that the additional disambiguation power of a solution $\mathcal{A}$ translates in practice, the estimate of the solution's learnability $\mathrm{Learnability}(\mathcal{A})$ is used to weigh improvements between $S_{\mathrm{oracle}}(\mathcal{A})$ and the accuracy of a system that predicts only according to the entity prediction model which is equivalent to the performance of having an empty solution : $S_{\mathrm{oracle}}(\emptyset)$.

Selecting a large number of type axes will provide strong disambiguation power, but may lead to degenerate solutions that are harder to train, slow down prediction, and lack higher-level concepts that provide similar accuracy with less axes. This is prevented using a per type axis penalty of λ.

Combining these three terms gives the equation for J :

$$J(\mathcal{A}) =(S_{\mathrm{oracle}}(\mathcal{A}) - S_{\mathrm{oracle}}(\emptyset) \cdot \mathrm{Learnability}(\mathcal{A}) - |\mathcal{A}| \cdot \lambda. \tag{7}$$

3.6 . Type Classifier

After the discrete optimization has completed a type system $\mathcal{A}$ is produced. This type system can now be used to label data in multiple languages from text snippets associated with the ontology [5], and supervize a Type classifier.

The goal for this classifier is to discover long-term dependencies in the input data that let it reliably predict types across many contexts and languages. For this reason a bidirectional-LSTM [83] is chosen as the neural network architecture. Robustness to spelling and casing variability is increased with the use of Word, prefix, and suffix embeddings as done in [6]. The network is shown pictorially in Figure III.4. The classifier is trained to minimize the negative log likelihood of the per-token types for each type axis i in the document D with L tokens, where the types given by type axis i for a token at position p is denoted $t_{i,p}$: $-\sum_{i=1}^{k} \log \mathbb{P}_i(t_{i,1}, \ldots, t_{i,L}|D)$. When using Wikipedia as the source of text snippets the label supervision is partial : labels are only found for tokens inside intra-wiki links. In order to model only the areas that have supervision, the objective is formulated with a conditional independence assumption :

$$-\log \mathbb{P}(t_{\forall_{i,j}}|D) = -\sum_{i=1}^{k}\sum_{j=1}^{L} \log \mathbb{P}_i(t_{i,j}|w_j, D). \tag{8}$$

5. Wikidata's ontology has cross-links with Wikipedia, IMDB, Discogs, MusicBrainz, and other encyclopaedias with snippets.

3.7 . Prediction

At prediction-time classifier belief is integrated in our decision process by first running it over the full context and obtaining a belief over each type axis for each input word $w_0, \ldots, w_L$. For each mention m covering words $w_x, \ldots, w_y$, a type conditional probability is produced for all type axes i : $\{\mathbb{P}_i(\cdot|w_x, D), \ldots, \mathbb{P}_i(\cdot|w_y, D)\}$. In multi-word mentions, beliefs must be combine over multiple tokens $x \ldots y$: the product of the beliefs over the mention's tokens is correct but numerically unstable and slightly less performant than max-over-time [6], which we denote for the i-th type axis : $\mathbb{P}_{i,*}(\cdot|m, D)$.

The score $s_{e,m,D,\mathcal{A},\theta} = \text{EntityScore}(e, m, D, \mathcal{A}, \theta)$ of an entity e given these conditional probability distributions $\mathbb{P}_{1,*}(\cdot|m, D), \ldots, \mathbb{P}_{k,*}(\cdot|m, D)$, and the entities' types in each axis $t_1, \ldots, t_k$ can then be combined to rank entities according to how predicted they were by both the entity prediction model and the type system. The chosen entity e^* for a mention m is chosen by taking the option that maximizes the score among the $\mathcal{E}_m$ possible entities ; the equation for scoring and e^* is given below, with α_i a per type axis smoothing parameter, β is a smoothing parameter over all types :

$$\mathbb{P}_{\text{Link}}(e|m) = \frac{\text{LinkCount}(m, e)}{\sum_{j \in \mathcal{E}_m} \text{LinkCount}(m, j)}, \tag{9}$$

$$s_{e,m,D,\mathcal{A},\theta} = \mathbb{P}_{\text{Link}}(e|m) \cdot \left(1 - \beta + \beta \cdot \left\{ \prod_{i=1}^{k} (1 - \alpha_i + \alpha_i \cdot \mathbb{P}_{i,*}(t_i|m, D)) \right\} \right). \tag{10}$$

4 . Results

4.1 . Training details and hyperparameters

4.1.1 Optimization

Our models are implemented in Tensorflow and optimized with Adam with a learning rate of 10^{-4}, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, annealed by 0.99 every 10,000 iterations.

To reduce over-fitting and make our system more robust to spelling changes we apply Dropout to input embeddings and augment our data with noise : swap input words with a special <UNK> word, remove capitalization or a trailing "s." In our Named Entity Recognition experiments we add Gaussian noise during training to the LSTM weights with $\sigma = 10^{-6}$.

We use early stopping in our Named Entity Recognition experiments when validation F1 score stops increasing. Type classification model selection is different

6. The choice of max-over-time is empirically motivated : we compared product mean, min, max, and found that max was comparable to mean, and slightly better than the alternatives.

as the models did not overfit, thus we instead stop training when no more improvements in F1 are observed on held-out type-training data (about 3 days on one Titan X Pascal).

Method	Parameter	Value
Greedy	b	1
Beam Search	b	8
CEM	N_{CEM}	1000
	p_{start}	$\frac{50}{\lvert\mathcal{R}\rvert} \approx 0.001$
	k_{CEM}	200
GA	G	200
	$N_{\text{population}}$	1000
	mutation probability	0.5
	crossover probability	0.2

TABLE III.1 – Hyperparameters for type system discovery search.

4.1.2 Neural Network Architecture

Character representation An effective way of representing unseen or rare words is to use character-aware representations of words. We apply this technique by using the neural network architecture from [77] where a convolutional neural network is applied to the input character embeddings with the following convolution filters (filter window size, filter channels) : {(1,50), (2, 75), (3, 75), (4, 100), (5, 200), (6, 200), (7, 200)}. Following the original architecture and hyperparameters, the character convolutions operate on words with a maximum length of 40 characters, and 15-dimensional character embeddings followed by 2 Highway layers [151]. We also learn 6-dimensional embeddings for 2 and 3 character prefixes and suffixes.

Step	Replacements	Links changed
1	1,109,408	9,212,321
2	13922	1,027,009
3	1229	364,500
4	153	40,488
5	74	25,094
6	4	1,498

TABLE III.2 – Link change statistics per iteration during English Wikipedia Anaphora Simplification.

Text Window Classifier The text window classifiers have 5-dimensional word embeddings, and use Dropout of 0.5. Empirically we find that two passes through the dataset with a batch size of 128 is sufficient for the window classifiers to converge. Additionally we train multiple type axes in a single batch, reaching a training speed of 2.5 type axes/second.

4.1.3 Data Preparation

Link statistics collected on large corpuses of entity mentions are extensively used in entity linking. These statistics provide a noisy estimate of the conditional probability of an entity e for a mention m $\mathbb{P}(e|m)$. Intra-wiki links in Wikipedia provide a multilingual and broad coverage source of links, however annotators often create link anaphoras : "king" $\rightarrow$ **Charles I of England**. This behavior increases polysemy ("king" mention has 974 associated entities) and distorts link frequencies ("queen" links to the band **Queen** 4920 times, **Elizabeth II** 1430 times, and **monarch** only 32 times).

Problems with link sparsity or anaphora were previously identified, however present solutions rely on pruning rare links and thus lose track of the original statistics [41, 58, 89]. We propose instead to detect anaphoras and recover the generic meaning through the Wikidata property graph : if a mention points to entities A and B, with A being more linked than B, and A is B's parent in the Wikidata property graph, then replace B with A. We define A to be the parent of B if they connect through a sequence of Wikidata properties {`instance of`, `subclass of`, `is a list of`}, or through a single edge in {`occupation`, `position held`, `series`[7]}. The simplification process is repeated until no more updates occur. This transformation reduces the number of associated entities for each mention ("king" senses drop from 974 to 143) and ensures that the semantics of multiple specific links are aggregated (number of "queen" links to *monarch* increase from 32 to 3553).

After simplification we find that the mean number of senses attached to polysemous mentions drops from 4.73 to 3.93, while over 10,670,910 links undergo

7. e.g. **Return of the Jedi** $\underset{\texttt{series}}{\rightarrow}$ **Star Wars**

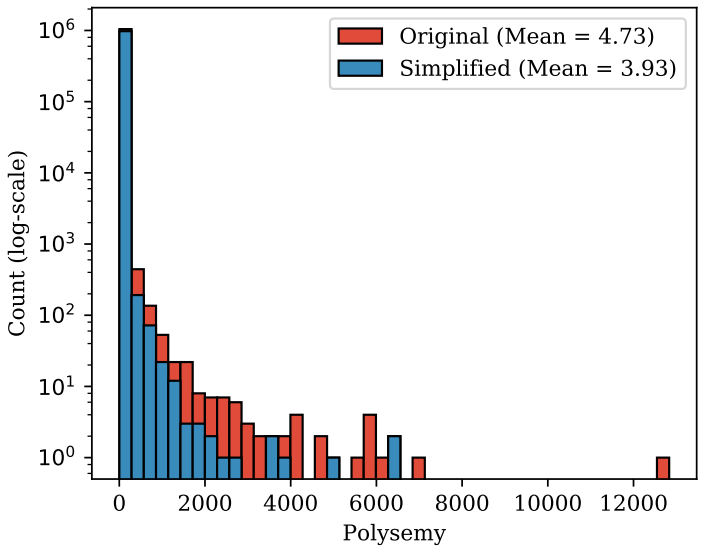


FIGURE III.5 – Mention Polysemy change after simplification.

changes in this process (Figure III.5). Table III.2 indicates that most changes result from mentions containing entities and their immediate parents. This simplification method strongly reduces the number of entities tied to each Wikipedia mention in an automatic fashion across multiple languages. We release the software for performing these link simplifications [8].

4.2 . Type System Discovery

In the following experiments we evaluate the behavior of different search methodologies for type system discovery : which method best scales to large numbers of types, achieves high accuracy on the target Entity Linking task, and whether the choice of search impacts learnability by a classifier or generalisability to held-out Entity Linking datasets.

For the following experiments we optimize DeepType's type system over a held-out set of 1000 randomly sampled articles taken from the February 2017 English Wikipedia dump, with the Learnability heuristic text window classifiers trained only on those articles. The type classifier is trained jointly on English and French articles, totalling 800 million tokens for training, 1 million tokens for validation, sampled equally from either language.

We restrict roots $\mathcal{R}$ and edges $\mathcal{G}$ to the most common $1.5 \cdot 10^5$ entities that are entity parents through `wikipedia category` or `instance of` edges, and eliminate type axes where $\mathrm{Learnability}(\cdot)$ is 0, leaving 53,626 type axes.

8. https://github.com/openai/deeptype

4.2.1 Human Type System Baseline

To isolate discrete optimization from system performance and gain perspective on the difficulty and nature of the type system design, a human-designed type system serves as a baseline. Human designers have access to the full set of entities and relations in Wikipedia and Wikidata, and compose different inheritance rules through Boolean algebra to obtain higher level concepts. For instance to define the concept of `woman` a membership rule which seeks the intersection of the descendants of "human" via the `instance of` relation, and the descendants of "female" via the `sex` relation. To construct a membership rule for `animal` which does not contain humans, we look for the descendants of "taxon" [9] via the `instance of` relation and exclude the descendants of "human" via the `instance of` relation. The final human system is given in Section 1 and uses 5 type axes [10], and 1218 inheritance rules.

To assist humans with the design of the system, the rules are built interactively in a REPL, and execute over the 24 million entities in under 10 seconds, allowing for real time feedback in the form of statistics or error analysis over an evaluation corpus. On the evaluation corpus, disambiguation mistakes can be grouped according to the ground truth type, allowing a per type error analysis to easily detect areas where more granularity would help.

4.2.2 Search methodologies

Beam Search and Greedy selection We iteratively construct a type system by choosing among all remaining type axes and evaluating whether the inclusion of a new type axis improves our objective : $J(\mathcal{A} \cup \{t_j\}) > J(\mathcal{A})$. The search uses a beam size of b and stops when all solutions stop growing.

Cross-Entropy Method (CEM) [138] is a stochastic optimization procedure applicable to the selection of types. The CEM search begins with a probability vector $\vec{P_0}$ set to p_{start}, and at each iteration N_{CEM} different vectors are sampled from the Bernoulli distribution given by $\vec{P_i}$. We denote a sampled vector s_t. Each sampled vector's fitness is measured using (7). The k_{CEM} highest fitness vectors form the winning population $\mathcal{S}_t$ at iteration t. The probabilities are fit to $\mathcal{S}_t$ giving $P_{t+1} = \frac{\sum_{\vec{s} \in \mathcal{S}_t} \vec{s}}{k_{\text{CEM}}}$. The optimization is complete when the probability vector is binary.

9. Taxon is the general parent of living items in Wikidata.
10. `IsA`, `Topic`, `Location`, `Continent`, and `Time`.

Genetic Algorithm The best subset of type axes can be found by representing type axes as genes carried by $N_{\text{population}}$ individuals in a population undergoing mutations and crossovers [57] over G generations. Individuals are selected using (7) as the fitness function.

4.2.3 Search Methodology Performance Impact

Approach	Evals	Accuracy	Items
BeamSearch	$5.12 \cdot 10^7$	97.84	130
Greedy	$6.40 \cdot 10^6$	97.83	130
GA	116,000	96.959	128
CEM	43,000	96.26	89
Random	N/A	92.9 ± 0.28	128
No types	0	92.10	0

TABLE III.3 – Type system discovery method comparison

Model	CoNLL 2003		OntoNotes	
	Dev	Test	Dev	Test
Bi-LSTM [21]	-	76.29	-	77.77
Bi-LSTM-CNN + emb + lex [21]	**94.31**	**91.62**	84.57	**86.28**
Bi-LSTM (Ours)	89.49	83.40	82.75	81.03
Bi-LSTM-CNN (Ours)	90.54	84.74	83.17	82.35
Bi-LSTM-CNN (Ours) + types	93.54	88.67	**85.11**	83.12

TABLE III.4 – Named Entity Recognition F1 score comparison for DeepType pretraining vs. baselines. Best results shown in bold.

4.3 . Effect of System Size Penalty

To validate that the hyperparameter λ (Subsection 3.5) controls type system size, and find the best tradeoff between size and accuracy, we experiment with a range of values and find that accuracy grows more slowly below 0.00007, while system size still increases. The effect averaged on 10 trials for a variety of λ penalties is shown in Figure III.8. In particular there is a crossover point in the performance characteristics when selecting λ, where a looser penalty has diminishing returns in accuracy around $\lambda = 10^{-4}$.

From this point on $\lambda = 0.00007$, and we compare the number of iterations needed by different search methods to converge, against two baselines : the empty set and the mean performance of 100 randomly sampled sets of 128 types (Table III.3). We observe that the performance of stochastic optimizers GA and CEM is similar to heuristic search, but requires orders of magnitude less function evaluations.

Type System size penalty λ vs. Solution Size

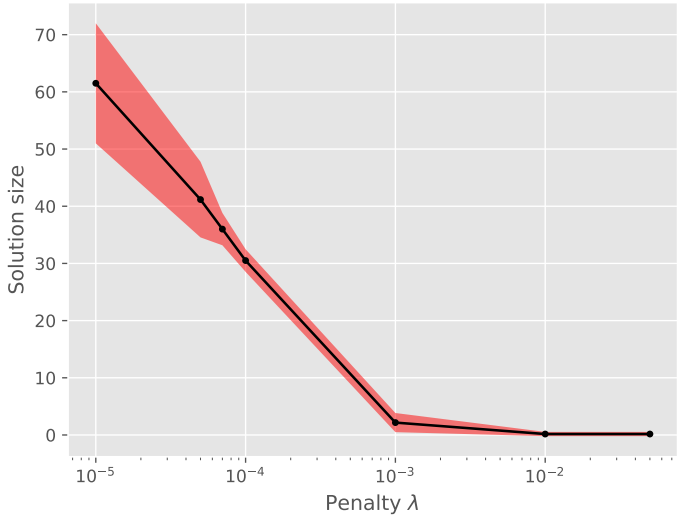


FIGURE III.6 – The size of the solution decreases exponentially with increased penalty (Standard deviation across 3 seeds shown in red).

Type System size penalty λ vs. Convergence Iterations

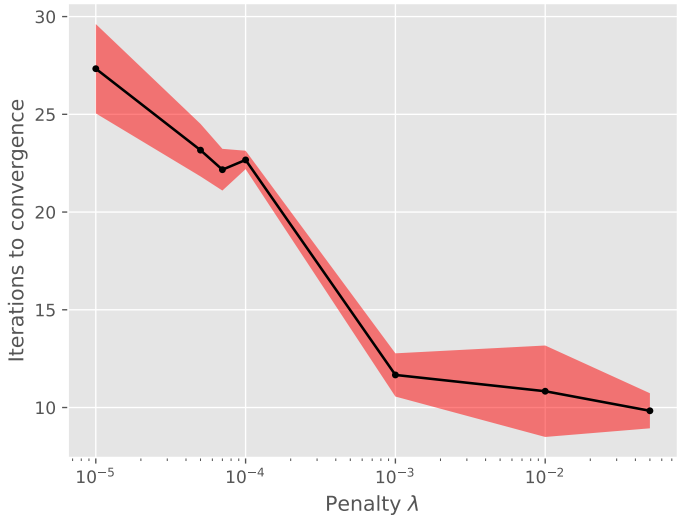


FIGURE III.7 – Systems with higher penalties require less iterations to converge (Standard deviation across 3 seeds shown in red).

Type System size penalty λ vs. Oracle Accuracy

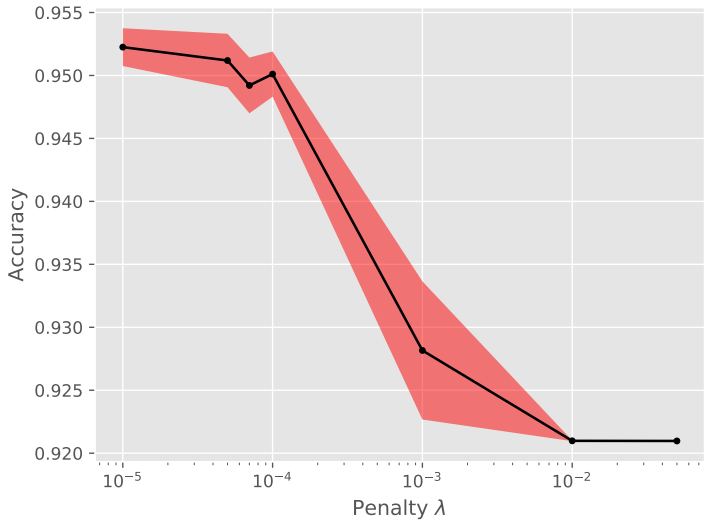


FIGURE III.8 – Accuracy increases with reductions in penalty, and plateaus near $\lambda = 10^{-4}$ (Standard deviation across 3 seeds shown in red).

Type System size penalty λ vs. Type System Objective J

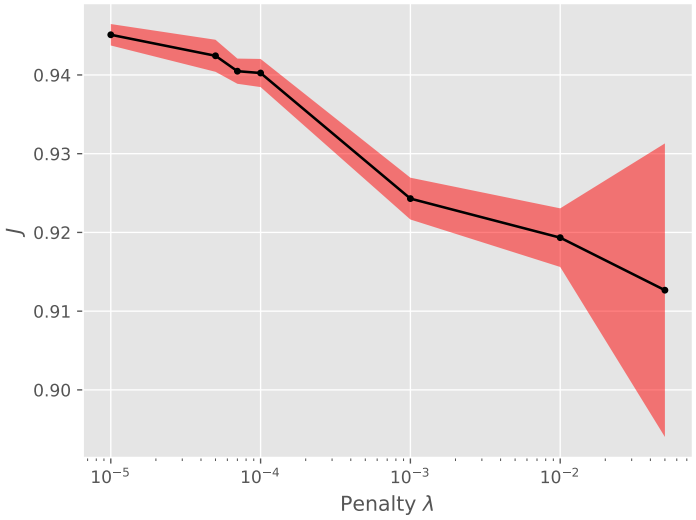


FIGURE III.9 – Objective J increases as penalty decreases, as the solution size is less penalized (Standard deviation across 3 seeds shown in red).

Model	Wiki				WKD	AIDA	TAC
	en	fr	de	es			
M&W [101]					84.6		
TagMe [41]	83.2		80.7		90.9		
Globerson et al. [49]						91.7	87.2
Yamada et al. [175]						91.5	85.2
NTEE [176]							87.7
LinkCount only	89.1	92.0	92.0	90.0	82.7	68.6	81.5
Human Oracle [124]						96.78	96.86
manual	**94.3****	93.0			91.9**	93.1**	90.7*
manual (oracle)	97.7	98.0	98.6	98.2	95.9	98.2	98.6
greedy	93.7**	**93.0**			**92.4****	94.2**	**90.9***
greedy (oracle)	98.0	97.2	97.9	98.3	97.3	99.0	98.3
CEM	93.7**	92.4			92.3**	94.0**	90.3*
CEM (oracle)	97.5	96.7	97.5	97.6	96.5	99.0	96.8
GA	93.7**	92.0			92.1**	**94.9****	90.3*
GA (oracle)	97.3	96.8	97.4	97.6	96.3	98.5	96.7
GA (English)	93.0**				91.7**	93.7**	

TABLE III.5 – Entity Linking model Comparison. Significant improvements over prior work (2018) denoted by * for $p < 0.05$, and ** for $p < 0.01$. Best non-oracle results shown in bold.

Next, we compare the behavior of the different search methods to a human designed system and state of the art approaches on three standard datasets (i.e. WIKI-DISAMB30 (WKD) [41] [11], AIDA [64], and TAC KBP 2010 (TAC) [69]), along with test sets built by randomly sampling 1000 articles from Wikipedia's February 2017 dump in English, French, German, and Spanish which were excluded from training the classifiers. Table III.5 has Oracle performance for the different search methods on the test sets, where we report disambiguation accuracy per annotation. A LinkCount baseline is included that selects the mention's most frequently linked entity [12]. All search techniques' Oracle accuracy significantly improve over LinkCount, and achieve near perfect accuracy on all datasets (97-99%) ; furthermore we notice that performance between the held-out Wikipedia sets and standard datasets sets is similar, supporting the claim that the discovered type systems generalize well. We note that machine discovered type systems outperform human designed systems : CEM beats the human type system on English

11. We apply the preprocessing and link pruning as [41] to ensure the comparison is fair.

12. Note that LinkCount accuracy is stronger than the one found in [41] or [101] because newer Wikipedia dumps improve link coverage and reduce link distribution noisiness.

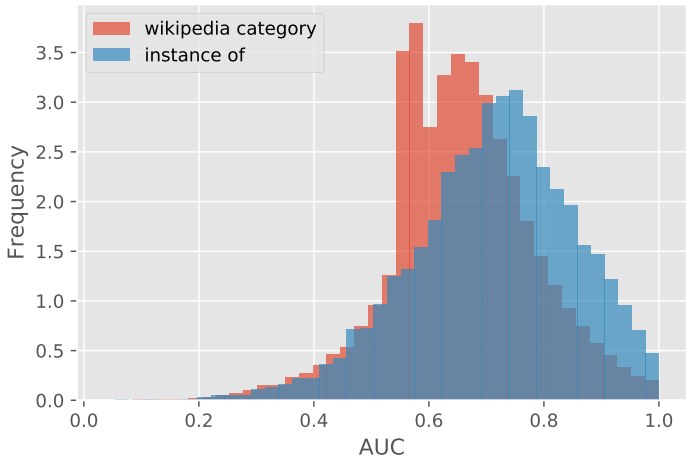


FIGURE III.10 – We plot the learnability score (AUC) for types derived from “instance of" and “wikipedia category" inheritance relations in the knowledge graph. Most “instance of" type-axes have higher AUC scores those from type axes produced using “wikipedia category".

Wikipedia, and all search method's type systems outperform human systems on WIKI-DISAMB30, AIDA, and TAC.

4.4 . Learnability Heuristic behavior

To better understand the behavior of the population of classifiers used to obtain AUC scores for the Learnability heuristic we investigate whether certain type axes are systematically easier or harder to predict. We find that type axes with a `instance of` edge have on average higher AUC scores than type axes relying on `wikipedia category` as visible in Figure III.10.

Furthermore, we also wanted to ensure that our methodology for estimating learnability was not flawed or if variance in our measurement was correlated with AUC for a type axis. We find that the AUC has low standard deviation as visible in the histogram in Figure III.11, and observe no relation between the standard deviation of the AUC scores for a type axis and the AUC score itself as visible in Figure III.12.

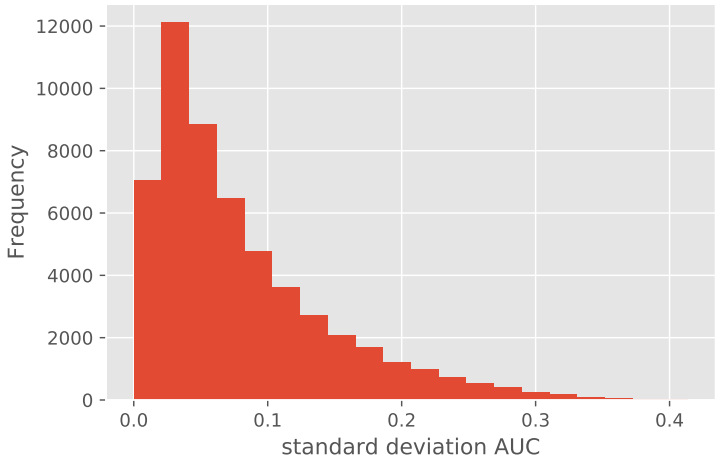


FIGURE III.11 – We construct a histogram of the standard deviation of the learnability score (AUC), and find that the standard deviation for AUC scoring with text window classifiers is below 0.1.

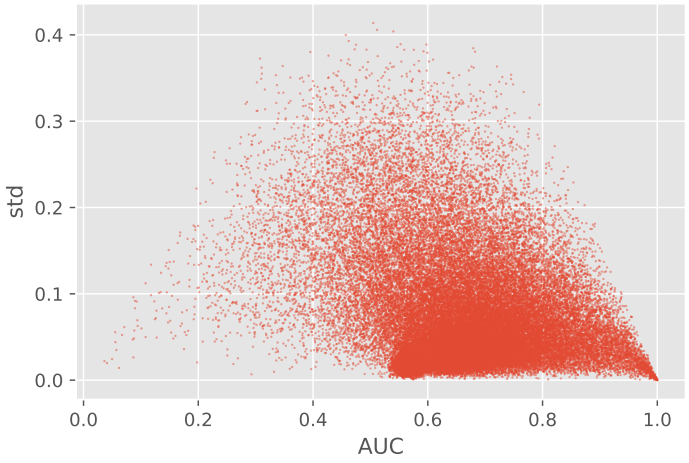


FIGURE III.12 – We plot the the learnability score (AUC) vs. the standard deviation of the learnability score (AUC) using 3 different random seeds to train the Learnability metric. AUC is not correlated with AUC's standard deviation.

4.4.1 Search Methodology Learnability Impact

To understand whether the type systems produced by different search methods can be trained similarly well we compare the type system built by GA, CEM, greedy, and the one constructed manually. Entity Linking Disambiguation accuracy is shown in Table III.5, where we compare with recent deep-learning based approaches [49], or recent work by Yamada *et al.* for embedding word and entities [175], or documents and entities [176], along with count and coherence based techniques Tagme [41] and Milne & Witten [101]. To obtain Tagme's Feb. 2017 Wikipedia accuracy we query the public web API [13] available in German and English, while other methods can be compared on CoNLL(YAGO) and TAC KBP 2010. Models trained on a human type system outperform all previous approaches to entity linking, while type systems discovered by machines lead to even higher performance on all datasets except English Wikipedia.

4.5 . Multilingual Transfer

4.5.1 Entity Linking Transfer

Type systems are defined over Wikidata/Wikipedia, a multi-lingual knowledge base/encyclopaedia, thus type axes are language independent and can produce cross-lingual supervision. To verify whether this cross-lingual ability is useful we train a type system on an English dataset and verify whether it can successfully supervize French data. We also measure using the Oracle (performance upper bound) whether the type system is useful in Spanish or German. Oracle performance across multiple languages does not appear to degrade when transferring to other languages (Table III.5). We also notice that training in French with an English type system still yields improvements over $\mathrm{LinkCount}$ for CEM, greedy, and human systems.

Because multi-lingual training might oversubscribe the model, we verified if monolingual would outperform bilingual training : we compare GA in English + French with only English (Row " GA (English)" in Table III.5). Bilingual training does not appear to hurt, and might in fact be helpful.

4.5.2 Multilingual Representation

We follow-up by inspecting whether the bilingual word vector space led to shared representations. Multilingual data creation is a side-effect of the ontology-based automatic labeling scheme. In Table III.6 we present nearest-neighbor words for words in multiple languages. We note that common words (he, Argentinian, hypothesis) remain close to their foreign language counterpart, while proper nouns group with country/language-specific terms (French and US politicians cluster separately).

We hypothesize that common words, by not fulfilling a role as a label, can therefore operate in a language independent way to inform the context of types,

13. `https://tagme.d4science.org/tagme/`

Word	k		
	1	2	3
Argentinian	argentin (0.259)	Argentina (0.313)	Argentine (0.315)
lui	he (0.333)	il (0.360)	him (0.398)
Sarkozy	Bayron (0.395)	Peillon (0.409)	Montebourg (0.419)
Clinton	Reagan (0.413)	Trump (0.441)	Cheney (0.495)
hypothesis	paradox (0.388)	Hypothesis (0.459)	hypothèse (0.497)
feu	killing (0.585)	terrible (0.601)	beings (0.618)
computer	Computer (0.384)	computers (0.446)	informatique (0.457)

TABLE III.6 – Top-k Nearest neighbors (cosine distance) in shared English-French word vector space.

while proper nouns will have different type requirements based on their labels, and thus will not converge to the same representation.

4.5.3 Multilingual Part of Speech Tagging

This	is	a	**car**	,	ceci	n'	est	pas	une	voiture	**car**	c'	est	un	**car**	.
PRO	VRB	DET	**NON**	PCT	PRO	PRT	VRB	ADV	DET	NON	**CON**	PRO	VRB	DET	**NON**	PCT

FIGURE III.13 – Model trained jointly on monolingual Part of Speech corpora detecting the multiple meanings of "car" (shown in bold) in a mixed English-French sentence. Words shown on the first row, and part-of-speech tags shown in the second row.

Finally the usage of multilingual allows some amount of subjective experiments. For instance below we show some samples from the model trained jointly on english and french correctly detecting the meaning of the word "car" across three possible meanings :

4.6 . Named Entity Recognition Transfer

The goal of our Named Entity Recognition experiment is to verify whether DeepType produces a type sensitive language representation useful for transfer to other downstream tasks. To measure this we pre-train a type classifier with a character-CNN and word embeddings as inputs, following [76], and replace the output layer with a linear-chain CRF [83] to fine-tune to Named Entity Recognition data. Our model's F1 scores when transferring to the CoNLL 2003 Named Entity Recognition task and OntoNotes 5.0 (CoNLL 2012) split are given in Table III.4. We compare with two baselines that share the architecture but are not pre-trained, along with the state of the art at the time of DeepType's publication [21].

We see positive transfer on Ontonotes and CoNLL : our baseline Bi-LSTM

strongly outperforms [21]'s baseline, while pre-training gives an additional 3-4 F1 points, with our best model outperforming [21] on the OntoNotes development split. While our baseline LSTM-CRF performs better than in the literature, our strongest baseline (CNN+LSTM+CRF) does not match the state of the art with a lexicon. We find that DeepType always improves over baselines and partially recovers lexicon performance gains, but does not fully replace lexicons.

5 . Discussion

5.1 . Overview

In this chapter we introduced DeepType, a method for integrating symbolic knowledge into the reasoning process of a neural network. We've proposed a mixed integer reformulation for jointly designing type systems and training a classifier for a target task, and empirically validated that when this technique is applied to Entity Linking it is effective at integrating symbolic information in the neural network reasoning process. When pre-training with DeepType for Named Entity Recognition, we observe improved performance over baselines and improvements over the 2017 state of the art on the OntoNotes dev set, suggesting there is cross-domain transfer : symbolic information is incorporated in the neural network's distributed representation. Furthermore we find that type systems designed by machines outperform those designed by humans on three benchmark datasets, which is attributable to incorporating learnability and target task performance goals within the design process. Our approach naturally enables multilingual training, and our experiments show that bilingual training improves over monolingual, and type systems optimized for English operate at similar accuracies in French, German, and Spanish, supporting the claim that the type system optimization leads to the discovery of high level cross-lingual concepts useful for knowledge representation. We compare different search techniques, and observe that stochastic optimization has comparable performance to heuristic search, but with orders of magnitude less objective function evaluations.

The main contributions are a joint formulation for designing and integrating symbolic information into neural networks, that enable us to constrain the outputs to obey symbolic structure, and an approach to Entity Linking that uses type constraints. Our approach reduces Entity Linking resolution complexity for a document with N mentions from $O(N^2)$ to $O(N)$, while allowing new entities to be incorporated without retraining, and we find on three standard datasets (WikiDisamb30, CoNLL (YAGO), TAC KBP 2010) that our approach outperforms all prior solutions by a wide margin, including approaches that rely on a human-designed type system [89] and the more recent work by Yamada et al. for embedding words and entities [175], or document and entities [176]. As a result of our experiments, we observe that disambiguation accuracy using Oracles reaches 99.0% on CoNLL (YAGO) and 98.6% on TAC KBP 2010, suggesting that Entity Linking would be

almost solved if we can close the gap between type classifiers and the Oracle.

5.2 . Future Work

These results suggest many directions for future research : could DeepType be applied to other problems where incorporating symbolic structure is beneficial ? Would additional expressivity in the type system such as hierarchy help close the gap between trained model and Oracle accuracy ? Are there benefits to relaxing the type classifier's conditional independence assumption ?

5.3 . Relation to Thesis

In the following chapter (Chapter IV), we show how we can build upon the results presented here. First, we remark that the use of types enabled superior performance while increasing transparency, reducing parameters, and improving sample efficiency. However, the use of human annotation for type system design and the use of a proxy objective make the system more prone to error and misalignment. By relaxing the conditional independence assumption, we show how we can eliminate human annotation, and use the disambiguation objective directly to train the entire system.

CHAPTER IV
End-To-End Type Reasoning

1 . Introduction

Breakthroughs in Natural Language Understanding from high-capacity language models with mask-based losses [32] and pre-training on web-sized corpuses [123] have produced a massive shift in the number of examples needed to tackle NLP tasks thanks to finetuning and world-knowledge pre-encoded in model weights. Entity linking similarly benefited from this wave of pre-trained language models [42, 87, 93] where systems without task-specific features match the accuracy of those with Entity Linking features and structured data [126, 146]. Despite advances from novel architectures and pre-training, Entity Linking systems fall short of human performance with accuracies ranging from 90% to 96% on standard benchmark datasets [87, 126], while other NLP tasks such as sentiment-analysis [123], named entity recognition [177], or part of speech tagging rival human performance with accuracies above 97%.

Have we reached a performance ceiling on Entity Linking ? We split this question into two parts : what is human performance on this task, and can we match it ?

We answer the first question in our human performance benchmark in Chapter II where we find a gap between algorithmic and human performance. In this chapter we answer the second question through our key contribution : DeepType 2.

DeepType 2 is an Entity Linking system that improves over the state of art (SoTA) on seven standard Entity Linking datasets and attains higher than human accuracy from our benchmark on TAC and AIDA. Most of our gains are explained by type interactions : an entity representation that captures rich inter-entity relations by encoding entities using their typed Wikidata neighbors. Predictions are coherent thanks to a document-wide score trained by a contrastive loss ; the score retains type-system's explanatory power by capturing the per-type contribution to each prediction. The system also enables practical use of document coherency features by materializing them on-the-fly during search with a knowledge base in the loop.

This chapter is structured as follows : Section 1 describes our approach ; Section 2 presents experiments that show how DeepType 2 profits from type-based representations, negative sampling, and global normalization ; Section 3 contains a discussion of the results and future work directions.

2 . Approach

2.1 . Neural Network Architecture

DeepType 2 uses a neural network that takes as input entire documents with their mentions. An illustration of the architecture is given in Figure IV.1.

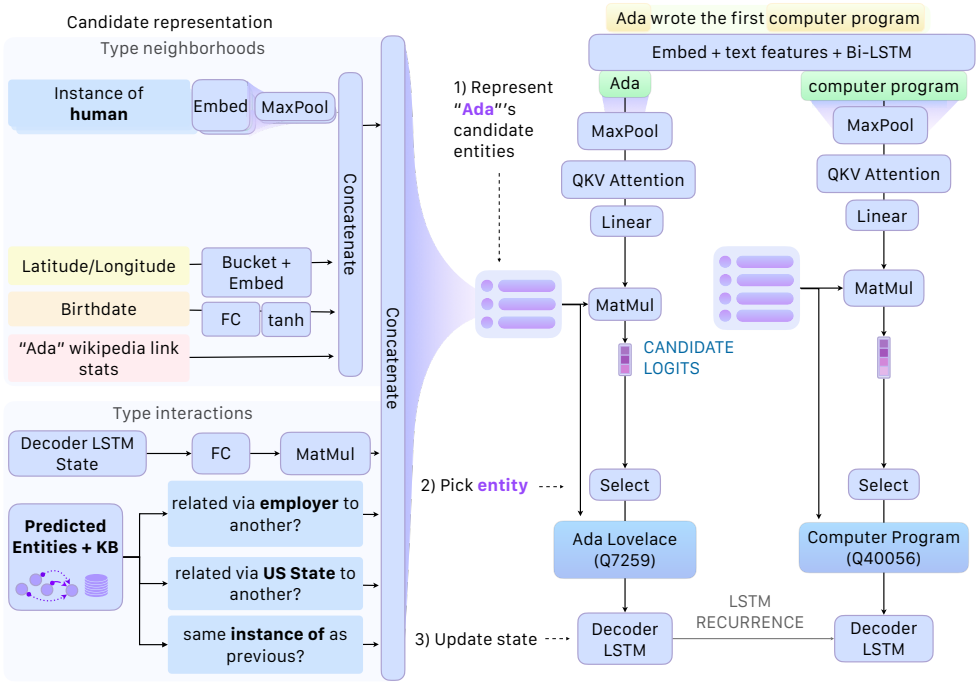


FIGURE IV.1 – An LSTM reads text, while a separate graph NN produces candidate entity representations used for prediction. Entity predictions are fed to a Decoder LSTM. The decoder LSTM and predicted entities produce type interaction features for future predictions.

2.1.1 Document Representation

The tokenized input document D is represented using word, prefix, and suffix embeddings and a capitalization bit. Tokens are processed by a stacked bidirectional-Long Short Term Memory (LSTM) RNN [51].

2.1.2 Mention Representation

For each mention an *alias table* generates candidate entities. The alias table is generated using the same approach as prior work [41] : intra-wiki links from Wikipedia provide a mapping from mention to linked entities. The link statistic features from the alias table are also exploited : 1) prior probability of linking to a particular entity given a particular alias table entry, 2) prior probability a given mention was seen for a given entity.

For each document-mention pair D_m a fixed-length representation $h_m(D_m)$ is obtained from the variable number of mention tokens. First a max-pool operation is applied on the associated Bi-LSTM hidden states to produce $h_{\text{pool},m}(D_m)$. Second, longer-range context for this set of tokens is obtained using QKV Attention [157] $(\text{Att}(\cdot))$ over the full document with $h_{\text{pool},m}(D_m)$ as query. The result of the attention operation is linearly projected into the same space as the max-pooled

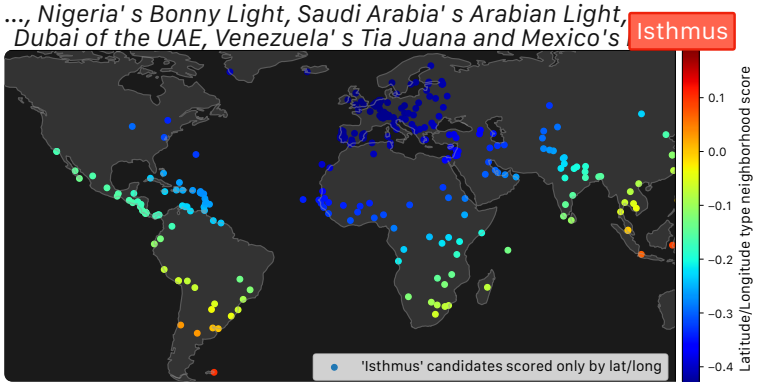


FIGURE IV.2 – By spoofing "Isthmus" candidates across the globe we observe the lat/long feature score grows in the Southern hemisphere.

and attended vector into the same space via a learnt matrix W to obtain h_{m} :

$$h_{\mathrm{m}}(D_m) = W \cdot h_{\mathrm{pool},m}(D_m) + \mathrm{Att}(h_{\mathrm{pool},m}(D_m)). \quad (1)$$

2.1.3 Entity Representation

Neighborhood relation	Wikidata relations
Admin. territorial entity	P131
Instance/Subclass of	P31, P279
Occupation	P106
Country	P27, P17, P495
Sport/Industry	P101, P425, P1995, P641, P2578, P452
Continent	P30
Gender	P21
Lat/Long	P625
Birthdate	P569, P571, P585, P580, P577

TABLE IV.1 – Wikidata relations for each type neighborhood.

Next, we associate to each candidate entity multiple sets of Wikidata neighbors (e.g. human, United Kingdom, mathematician) coming from different typed relations such as *occupation* or *origin country*. These neighborhood relations are chosen based on usage frequency in Wikidata. See Table IV.1 for the list of type neighborhood relations. The neighbors obtained from these relations can be entities, real values (e.g. latitude/longitude), or dates (e.g. birthdate).

We refer to neighbors that are up to n_{depth} steps away as the *type neighborhood* representation of an entity. See Figure IV.3 for an example of Ada Lovelace's type neighborhood.

To recover a fixed length entity representation from the type neighborhood we use a Graph Neural Network (GNN). In this work $n_{depth} = 2$, which enables us to

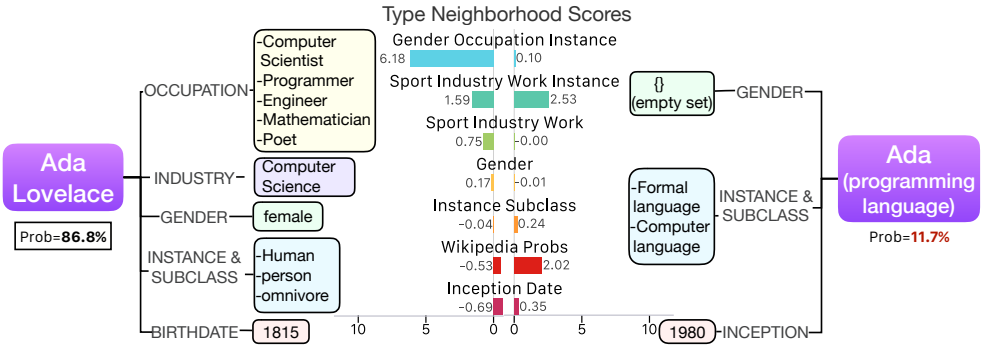


FIGURE IV.3 – Disambiguating "Ada" in the sentence "Ada wrote the first computer program. She..." Type neighborhoods for candidate entities are computed by finding depth 2 neighbors via different typed Wikidata edges. An entity's score is the sum of its type neighborhood and interaction scores. This acts as a rationale for DeepType 2's decisions. We see wikipedia probs, gender, occupation, instance, and work had the largest impact.

take advantage of a basic GNN consisting of an embedding layer and a max-pool. For deeper neighborhoods, a depth or edge-aware GNN might be preferable [163].

2.1.4 Type Interactions

We perform joint predictions over all mentions in a document. In order to do this, we augment the entity representation with two sets of features related to past predictions : latent and discrete type interactions.

Latent type interactions are obtained by computing the scalar product between the type neighborhood representation of a candidate and the hidden state of a decoder LSTM (3 in Figure IV.1). The decoder LSTM receives as input the chosen entity's type neighborhood representation after each prediction. Latent interactions measure if the candidate's type neighborhoods match the memory using a learnt function.

Discrete type interactions are boolean features corresponding to the result of multiple knowledge graph queries. For each relation in a prefined set, a knowledge graph query checks if any past entity is connected to the candidate entity by this relation. Using these features it is possible to measure list type-homogeneity or answer questions such as "is this candidate of the same sport / team / league/ etc. as past entities ?" As with type neighborhoods, relations were chosen based on their Wikidata usage frequency. As we later discuss in Section 3.3.2, certain relations are redundant, and the system is robust to removing those. See Table IV.2 for the list of Wikidata relations used in the type interactions. The discrete interactions access outside information from a KB to answer factual inter-entity questions. We provide in Figure IV.4 an example of these interactions to disambiguate John Gorst.

Type Interaction	Entity relation
Identity	same entity
League	P118
Season	P5138
Educated at	P69
Political Party	P102
Spouse	P26
Sibling	P3373
Employer	P108
Member of sports team	P54
Sport	(sport) P641, (occupation) P106, (field of this occupation) P425 and connective node inherits from Q31629 (sport).
US State	P131 and connective node inherits from Q35657.
Contemporary	overlap in (birthdate P569, deathdate P570).

TABLE IV.2 – Wikidata relations for each type interaction.

2.1.5 Scoring

Candidate probabilities are obtained from the dot product between the mention and the entity representation : with $c_0, \ldots, c_n$ candidate entities, s the discrete/latent state, and $f_{\mathrm{t}}(c_i, D_m, s)$ the concatenation of type neighborhood and F different interaction features :

$$\mathrm{Score}(c_i, D_m, s) = h_{\mathrm{m}}(D_m) \cdot f_{\mathrm{t}}(c_i, D_m, s), \tag{2}$$

$$\mathbb{P}(c_i | D_m, s) = \frac{\exp\left(\mathrm{Score}(c_i, D_m, s)\right)}{\sum_{j=0}^{n} \exp\left(\mathrm{Score}(c_j, D_m, s)\right)}. \tag{3}$$

The feature-vector $\mathrm{Score}(c_i, D_m, s)$ is formed by concatenating the F interaction features : $I_0(c_i), \ldots, I_F(c_i)$. Based on the (arbitrary but fixed) concatenation order feature $I_j(c_i)$ will be elementwise multiplied with a different subset of the vector h_m. Let us define $\mathcal{I}_{I_j(c_i)}$ to be the set dimensions of h_m that will be element-wise multiplied with $I_j(c_i)$, and $h_{m,z}$ to scalar at dimension z of h_m, then, we can recover feature scores as follows :

$$\mathrm{Score}(c_i, D_m, s) = \sum_{j=0}^{F} \underbrace{(h_{\mathrm{m},z \in \mathcal{I}_{I_j(c_i)}}(D_m)) \cdot I_j(c_i, s))}_{\text{feature } I_j\text{'s score}}, \tag{4}$$

$$= \sum_{j=0}^{F} \mathrm{Score}_{I_j}(c_i, D_m, s). \tag{5}$$

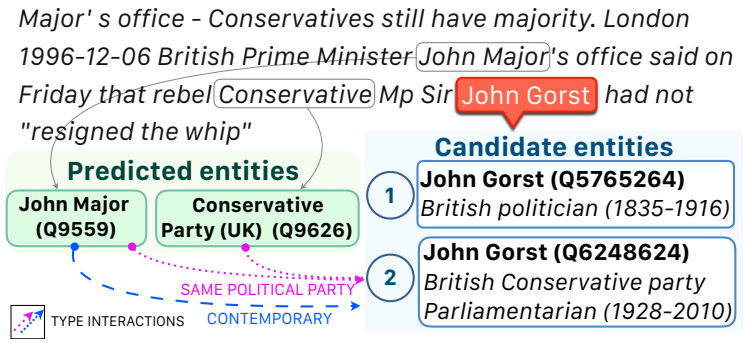


FIGURE IV.4 – In AIDA, type interactions with past predictions give us hints about "John Gorst"'s candidate entities : candidate 2 is contemporary to John Major and his political party is previously mentioned.

Rewriting $\text{Score}(c_i, D_m, s)$ as a sum of feature scores reveals each type neighborhood or interaction's contribution to the overall score. Feature scores may serve as decision justifications as we show in Figure IV.3 and Figure IV.2.

2.1.6 Objective Function

Model parameters θ are learnt by minimizing $\mathcal{L}(\theta)$, the negative log likelihood of the ground truth entity e relative to alias table candidates for the mention m :

$$\mathcal{L}(\theta) = \sum_{\{e, D_m, s\}} -\log \mathbb{P}(e|D_m.s; \theta). \tag{6}$$

2.2 . Contrastive Loss

Our objective function profits from becoming a contrastive loss. When too many candidates are returned by the alias table we subsample to reduce computational cost, and when there are too few, we supply negative samples [55]. Negative samples massively increase the supervision signal as over 45.4% of Wikipedia mentions are unambiguous.

A further reason to use a contrastive loss is its ability to focus model capacity towards only resolving actual ambiguities from the alias table. By comparison, a generative loss for predicting types independently wastes capacity on modeling all type combinations (e.g. about 2^{128} in DeepType [126]) most of which are impossible. A contrastive loss focuses the learning on discriminative features : the gradient is zero for features common between candidates (see proof of Lemma 1). Indeed, the likelihood of a candidate entity is computed by computing an exponential normalization (Softmax) over the scores of all candidates. Because exponential normalization is shift invariant, a feature that is common across multiple candidates is a constant that can be factorized and removed.

Lemma 1. *Given candidates $c_0, \ldots, c_n$, represented by features $I_0(c_i), \ldots, I_j(c_i)$, and the probability of a candidate c_i defined by $\mathbb{P}(c_i|D_m, s) \propto \exp(\sum_{j=0}^{F} \text{Score}_{I_j(c_i)})$,*

then if feature I_j is equal for all candidates, $I_j(c_i) = I_j(c_k) \forall_{(i,k)\in[0,n]}$, then $\nabla_{I_j}\mathbb{P}(c_i|D_m, s) = 0$.

Proof. Consider candidates $c_0, \ldots, c_n$ sharing a common type neighborhood or interaction feature I_k, making all type scores are equal to a constant C :

$$C = \text{Score}_{I_k(c_i)} = \cdots = \text{Score}_{I_k(c_n)}, \tag{7}$$

$$\tag{8}$$

then the feature I_k has 0 gradient as we can see by rewriting the probability of a candidate c_i using C :

$$\mathbb{P}(c_i|D_m, s) \propto \exp(\sum_{j=0}^{F} \text{Score}_{I_j(c_i)}), \tag{9}$$

$$\propto \exp(\sum_{j=0, j\neq k}^{F} \text{Score}_{I_j(c_i)}) \cdot \exp(C) \tag{10}$$

$$= \frac{\exp(\sum_{j=0,j\neq k}^{F} \text{Score}_{I_j(c_i)}) \cdot \exp(C)}{\left(\sum_{i=0}^{n} \exp(\sum_{j=0,j\neq k}^{F} \text{Score}_{I_j(c_i)})\right) \cdot \exp(C)} \tag{11}$$

$$= \frac{\exp(\sum_{j=0,j\neq k}^{F} \text{Score}_{I_j(c_i)}))}{\sum_{i=0}^{n} \exp(\sum_{j=0,j\neq k}^{F} \text{Score}_{I_j(c_i)})}. \tag{12}$$

Having eliminated I_k from the equation our result follows :

$$\nabla_{I_k}\mathbb{P}(c_i|D_m, s) = 0. \tag{13}$$

□

While the shift-invariance of Softmax is well known, it is however useful to note that this elimination of the gradient for I_k from our loss is thanks to exponential normalization. This property does not show up in a margin loss without normalization unless the score of negative samples is averaged.

2.3 . Densification

Property	Original	Dense
Number of Tokens	1.2B	2.8B
Number of Mentions	74M	220M
Paragraphs with 1+ Mentions	21.7M	68.6M

TABLE IV.3 – Wikipedia corpus densification statistics

In order to observe type interaction features we densify mentions in documents. For training, we densify Wikipedia articles by creating new links to entities already

present in the page. We filter new links with a classifier trained on 300 hand-collected labels. As articles do not refer to themselves, the subject of the article can be used to create many additional links. Keeping the high confidence new links increases dramatically the size of our training corpus by 2.97x from 74M to 220M mentions as detailed in Table IV.3. As some phrases are overly generic or ambiguous, we use a binary classifier to decide on new links. A similar technique was employed in DAWT to [150] to increase the number of links by 4.8x to obtain a mention detection and entity co-occurence dataset, but to the best of our knowledge we are the first to use this for Entity Linking.

2.3.1 Link Densification Network Architecture

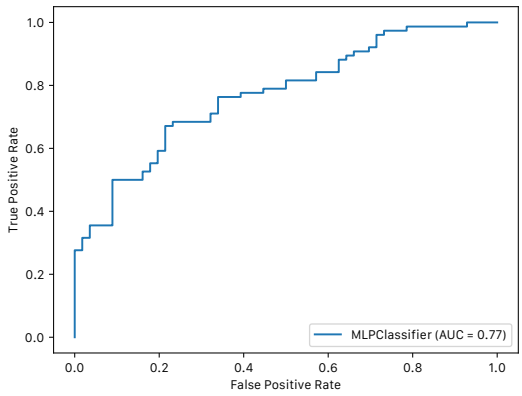


FIGURE IV.5 – ROC Curve for the synthetic link classifier.

The classifier is 2-layer Multi-Layer-Perceptron with 5 hidden units, a ReLu nonlinearity, using 400 manually collected labels with a 2 :1 train test split. and trained with the Adam optimizer [78]. The classifier has an accuracy of 71%, and its ROC curve is shown in Figure IV.5.

2.3.2 Distributional Shift from Densification

	Wikipedia negative log likelihood	
Training data	Original	Densified
Original	0.73	1.13
Densified	0.81	0.84

TABLE IV.4 – Impact of Wikipedia Densification on negative log likelihood.

To understand whether this densification produces data that prevents transfer to original Wikipedia data we look at the negative log likelihood of links when training with and without the densification in Table IV.4. We see a small change

in negative log likelihood when transferring to original Wikipedia data after training on the augmented set. A model trained only on the original set has a much greater increase in negative log likelihood when evaluated on densified data. From this investigation, we conclude that densification generalizes well to the original Wikipedia data. Densifying does not prevent a model from operating on infrequent mentions, but enables better handling of increased self-links. Conversely, the poor generalization to the densified data suggests that models trained purely on undensified Wikipedia data do not handle well an increase in mentions.

2.4 . Coherency

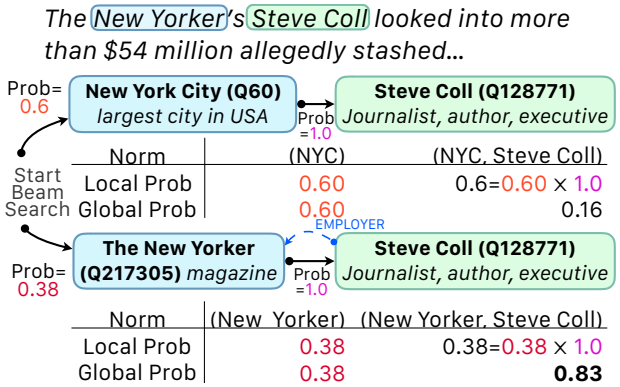


FIGURE IV.6 – Global normalization effect in TAC : Steve Coll, although unambiguous, reinforces the likelihood of picking his employer New Yorker magazine when scores are summed before being normalized.

Type interaction Syntax pattern	Entity relation
city/county , state/region	P131
city/county/state/region , country	P17
list-like : A **and** B or A,B, **and** C etc.	P31
human, (team)	P54
human, (nationality)	P495

TABLE IV.5 – Wikidata relations used within syntactic patterns.

To make coherent predictions we jointly predict entities while taking into account interactions between all predicted entities :

— Discrete type interactions act as constraints to prune the candidate search space : in the context of syntactic structures such as “Venice, California" we expect a located-in relation, or “Paris and London" where we expect list type homogeneity. These patterns are given in Table IV.5 with syntax in bold.

— Beam search with autoregressive features increases increases coherency with limited computational cost : DeepType 2's pairwise entity features are only

materialized during search. For k search beams, and document D with N_{D} input tokens, N_{m} mentions, M_{c} candidates per mention, and $N_{\mathrm{m}} \cdot N_{\mathrm{D}}$ attention features, the computational complexity is $O(N_{\mathrm{m}} \cdot M_{\mathrm{c}} \cdot k + N_{\mathrm{m}} \cdot N_{\mathrm{D}})$ instead of $O(N_{\mathrm{m}}^2 \cdot M_{\mathrm{c}}^2)$ if all features had to be pre-computed. The practical implications is that this system can process all AIDA with 16 search beams in 23s (187.3 mentions/s and 2178 tokens/s) on an NVIDIA GeForce GTX 1080.

— Global normalization enables every decision, regardless of order, to increase or decrease the joint likelihood of the prediction sequence. This is achieved by summing all decision scores before normalizing, rather than multiplying locally normalized probabilities as shown in Figure IV.6. This conversion from local to global was previously proposed to improve expressivity and overcome label bias [6, 125], an autoregressive model pitfall [82].

3 . Results

We evaluate DeepType 2 on standard benchmarks and on the human benchmark we presented earlier in Chapter II. Second, we investigate through ablations what aspects of the proposed approach are the most important.

In all our experiments DeepType 2 is trained for 2 million gradient steps using as annotations intra-wiki links from the December 2017 English Wikipedia dump with densification, as well as AIDA's train split. Unless otherwise noted, we use 16 search beams and global normalization. Training takes approximately 6 days on a single NVIDIA GTX 1080Ti on a computer with 128GB of RAM and 28 core 3.3Ghz Intel i9 CPUs. To facilitate comparisons with prior work on AIDA we use the PPR4NED alias table [114], otherwise our alias table is built from intra-wiki links.

DeepType 2's neural network dimensions and learning rate schedule were selected using a Wikipedia-based validation set and are provided in Table IV.6. Type neighborhoods use the embedding dimensions in Table IV.7. We also construct type neighborhoods that are a combination of these neighborhoods in Table IV.8. To obtain a representation for combinations we concatenate the max-pooled result of the individual type neighborhoods and process them using a fully connected layer and a ReLu nonlinearity. Dimensions for these fully connected (FC) layers are given in Table IV.8.

We use the Adam optimizer [78]. We resize training batches to contain at most 12,800 tokens per batch. If an out of memory error occurs we sample another batch and keep training. We accumulate gradients across 2 mini batches to fit within GPU memory when training with 100 negative samples.

Hyperparameter name	Value
Input Bi-LSTM size	512
Input Bi-LSTM layers	2
Attention Heads	2
Attention Query size	128
Word embedding size	200
Word UNK Probability	0.25
Word Vocab Size	750,000
Word {Prefix,Suffix}-{2,3} embedding size	6
Word {Prefix,Suffix}-{2,3} vocab size	100,000
Wikipedia link stats Layer size	20
Wikipedia link stats Dropout probability	0.3
Wikipedia link stats power	0.18
Decoder LSTM size	128
Learning Rate	0.001
LR Decay/33,000 gradient steps	1%
LR Decay/400,000 gradient steps	80%
AIDA Train data oversampling	10
Negative Samples	100
Training max candidate entities	100

TABLE IV.6 – Neural Network Hyperameters

3.1 . Evaluation on Standard Datasets

We compare Human performance, DeepType 2, and the current Entity Linking state of the art on the standard benchmark datasets TAC and AIDA and report our results with average and standard deviation across 6 runs in Table IV.9. In Table IV.10 we report evaluations of our system on five additional well known Entity Linking datasets WNED-WIKI [54], WNED-CWEB [54], MSNBC [27], AQUAINT [101], and ACE 2004 [130].

DeepType 2 improves accuracy over the SoTA on all evaluated datasets, and outperforms the human oracle accuracy by 0.62% on TAC and 0.74% on AIDA. The largest gains relative to prior work are observed on TAC (**2.58%**), AIDA (**1.02%**), WNED-CWEB (**3.77%**), while the smallest is WNED-WIKI. (0.43%).

Neighborhood relation	Vocab size	Min count	d
Admin. territorial entity	17003	10	10
Instance/Subclass of	14624	5	40
Occupation	1421	10	10
Country	759	3	10
Sport/Industry	599	10	40
Continent	12	10	10
Gender	3	10	10

TABLE IV.7 – Type neighborhood used to represent entities.

Neighborhood relations	Merge FC Dimension
Gender, Occupation, Instance	20
Sport/Industry, Instance	20

TABLE IV.8 – Type neighborhoods with cross-terms.

3.1.1 Mention Densification

One of the largest gains relative to prior work is observed on TAC, greatly thanks to the way mention "densification" provides additional contextual entities that power type interaction : we add mentions to the document to increase their frequency from TAC's original single mention/document. Mentions are detected by greedily taking the longest alias table matches linkable to persons, places, or activities. Accuracy increases by **3.97%** from 93.51% to 97.48%.

3.1.2 Joint Decision Making

The score given to a sequence of predictions is heavily dependent on type interaction features to make coherent decisions. We report the result of independent predictions versus joint predictions in Table IV.11. We observe a massive improvement over independent decisions when jointly predicting entities. A smaller but noticeable improvement is visible when switching from locally to globally normalized scores.

We also study the effect of varying the number of search beams in Table IV.12. We find that a small percentage of search errors in TAC and AIDA can be mitigated by considering more hypotheses.

Model	TAC	AIDA
Human Oracle	96.86	96.78
DeepType 2 (ours)	**97.48** ±0.06	**97.72** ±0.04
Ling et al. [87]	89.8	94.9
Raiman and Raiman [126]	90.85	94.88
Mulang' et al. [104]	-	94.94
Févry et al. [42]	94.9	96.7

TABLE IV.9 – Humans and state of the art Entity Linking system accuracy ($\mu \pm \sigma$). Best results shown in bold.

Dataset	DeepType 2 (ours)	Yang et al. (2018)	De Cao et al. [29]
W-CWEB	**85.57**±0.24	81.8	77.3
W-WIKI	**87.83**±0.08	79.2	87.4
MSNBC	**95.12**±0.23	92.6	94.3
AQUAINT	**92.74**±0.27	89.9	89.9
ACE 2004	**92.23**±0.19	89.2	90.1

TABLE IV.10 – Entity Linking system accuracy on standard datasets ($\mu \pm \sigma$). Best results shown in bold.

Decision Method	TAC	AIDA
Independent	93.51±0.07	96.76±0.08
Joint Local Score	97.44±0.08	97.62±0.07
Joint Global Score	**97.48**±0.06	**97.72**±0.04

TABLE IV.11 – Impact ($\mu \pm \sigma$) of decision method on accuracy. Best results shown in bold.

3.2 . Error Analysis

DeepType 2 has the ground truth entity in its top-3 responses over 99% of the time (99.10% on TAC, 99.35% on AIDA). The main remaining mistakes made by DeepType 2 and humans fall into the same category : confusing places and sports teams due to journalistic shorthand overloading the meaning of place names as visible in Table IV.13.

k	TAC	AIDA
1	97.44±0.08	97.69±0.06
8	97.44±0.08	97.71±0.04
16	**97.48**±0.06	**97.72**±0.04

TABLE IV.12 – Impact ($\mu \pm \sigma$) of varying search beams k on accuracy. Best results shown in bold.

Confusion	TAC (%)		AIDA (%)	
	DT2	Human	DT2	Human
Place vs. Sports Team/Club	**22.2**	**32.6**	**8.9**	**20.2**
Business vs. Business	18.5	7.0	2.4	0.8
Ethnic group vs. Country	3.7	3.1	0.8	27.4
Sports team vs. Sports team	3.7	9.3	0.0	13.7
Remainder	51.9	48.1	37.5	37.9

TABLE IV.13 – Typed confusions for DeepType 2 (DT2) and humans. The biggests source of errors is shown in bold.

3.3 . Ablations

3.3.1 Entity Representation

The comparison of different entity representations in DeepType 2 shows that the best one uses both type neighborhoods and type interactions as visible in Table IV.14. We empirically verify the effect of replacing type neighborhoods by same dimension unique Entity-Vectors used in state of the art approaches [42, 87, 146, 175]. Type neighborhoods have 6 times less parameters (166M vs. 998M), get the same accuracy as Entity-Vectors after training on a 1/4 of data and 10 times less updates (150k vs. 1.5M), and reach higher accuracy model on TAC and AIDA. We also ablate the use of type interactions and find that they also contribute to a large portion of the Entity-Vectors system's performance.

Representation	TAC	AIDA
type neighborhoods + type interactions *	**97.48**	**97.72**
unique entity vector + type interactions	94.07	94.57
unique entity vector	89.60	92.73

*Our proposed approach.

TABLE IV.14 – Impact of entity representation on accuracy. Best results shown in bold.

3.3.2 Type Interaction Features

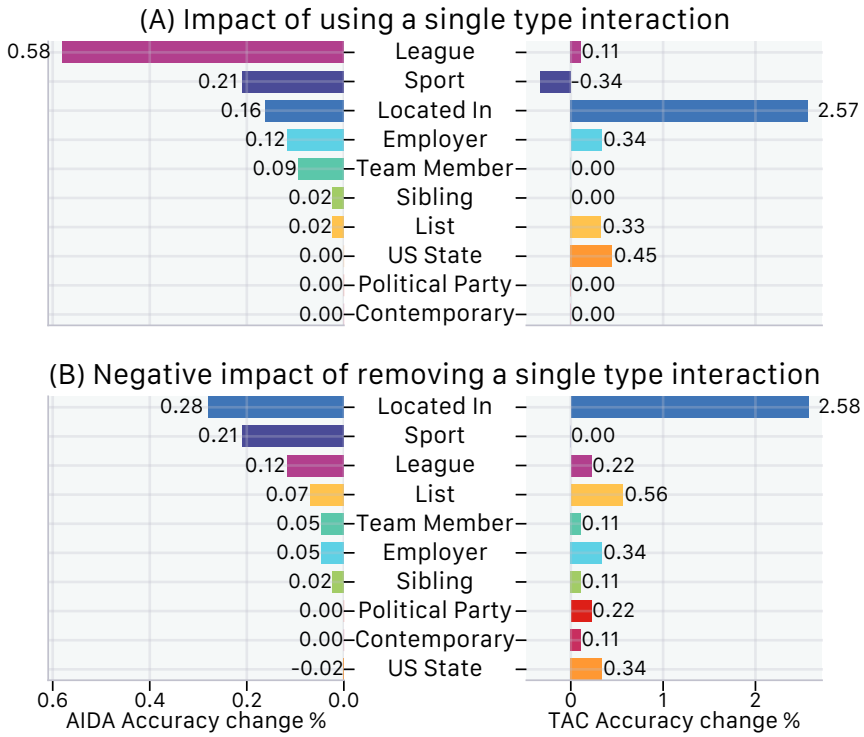


FIGURE IV.7 – Type interactions are domain dependent as visible in (A) by looking at the impact of using a single relation in TAC vs. AIDA. In (B) we test the redundancy of type interactions by removing one from the system.

Our entity representation ablation above shows type interactions are crucial, begging the question : what are the most important type interactions ? We compare the impact of using a single type interaction on TAC and AIDA accuracy in the (A) pyramid plot in Figure IV.7. We observe that type interactions are domain-dependent : relations such as “League" matter more in sports-heavy AIDA, and geographical relations (e.g. “Located in") benefit the newswire-based TAC.

As type interactions can have overlapping roles, we look at the sensitivity to removing a single type interaction as an indication of its redundancy and report the results in the (B) pyramid plot of Figure IV.7. “Located in" has the largest negative impact when removed and thus is least redundant. Conversely, “League" appears redundant as it individually increases AIDA accuracy by 0.58% accuracy, but only causes a 0.12% decrease if removed when all other type interactions are present.

3.4 . Training Ablations

Training data	TAC	AIDA
Original	96.75	96.47
Densified	**97.48**	**97.72**

TABLE IV.15 – Impact of Wikipedia Densification on accuracy. Best results shown in bold.

Negative Samples	Max entities per mention (training)	TAC	AIDA
0	20	95.41	95.63
20	20	96.98	**97.99**
100	100	**97.48**	97.72

TABLE IV.16 – Negative sampling impact on Entity Linking performance. Best results shown in bold.

3.4.1 Wikipedia Densification

We compare the quality of models trained with and without densification. With densification models obtain higher accuracy on TAC and AIDA as shown in Table IV.15.

3.4.2 Negative sampling

As entity representations are only learnt through comparisons, the unambiguous mentions provide no supervision potentially leaving representations untrained. In Table IV.16 we show that increased negative samples and training candidate entities improve final accuracy. Some negative samples are critical to performance, while increasing the number of training candidates from 20 to 100 is more helpful on TAC than AIDA.

4 . Discussion

4.1 . Overview

Through our human performance benchmark introduced in Chapter II, we observe that previous systems approach human performance but still underperform. We close the performance gap thanks to a new Entity Linking system, DeepType 2. The proposed approach removes the need for a pre-trained language model and improves over the human accuracy on the benchmark datasets and reaches a new state of the art on five other commonly used Entity Linking datasets.

The performance gains are explained by a novel abstract entity representation built on Wikidata relation subgraphs. Through ablations we show that this entity representation uses 80% fewer parameters than equivalent entity vectors,

and reaches higher accuracies thanks to an ability to share learning between entities of the same type. The strongest contributor to performance is the set of autoregressive relational features we call *type interactions*. These features enable the system to produce coherent document-wide predictions through higher order reasoning over the entity types (e.g. shared employers, geographical co-occurence, list type homogeneity). A further benefit of DeepType 2 is that it eliminates two major difficulties of existing type based systems such as DeepType [126] : 1) the type representation is now automatically generated by embedding subgraphs rather than curated type labels, 2) a single task-aligned objective function replaces prior use of a proxy multi-objective type classification.

4.2 . Future Work

The work presented in this chapter has several limitations. First, DeepType 2 relies solely on structured relations and cannot make use of the wealth of unstructured relations. Second, the presented system DeepType 2 does not take advantage of pre-trained language models. A useful line of investigation would be to test the effect of pre-training and alternate text encoding mechanisms.

4.3 . Relation to Thesis

The presented system reaches one of the thesis goals of attaining human level performance. However, the proposed approach can only access external knowledge using structured knowledge bases designed by humans. In the next chapter (Chapter V) I show how this system can be improved by combining DeepType 2 with a Pretrained Language Model to take advantage of unstructured relations.

The main strength of the system presented in this chapter is type interactions : these features enable the system to use external facts and reason about the full document context. This is greatly facilitated by the typing information associated with relations, enabling separation between uninformative relations, while providing shared features that a neural network can be trained to recognize when relevant relation types are present. Because unstructured relations lack this typing information, their inclusion may either hurt performance or be altogether ignored by the neural network. To overcome these difficulties, I propose to instead unify the representations of structured and unstructured relations : the neural network cannot distinguish the two sources of data, and must then learn to operate indifferently with either source present.

CHAPTER V

Neural Relational Database

1 . Introduction

The ability for large pretrained language models such as T5, PaLM, or GPT-3 [19, 23, 123] to implicitly recognize tasks using prompts has caused a massive shift in the way we think and design Natural Language Understanding systems that must change given the context or external facts.

Prior to prompt programming, the bulk of a task's definition and relevant facts were baked into the model during training. A model had to be retrained to alter its behavior, or built with externally controllable features informing the task such as gazeteers [94] or databases [165]. The reliance on external resources added resiliency and longevity by providing ways to perform updates and have the model react accordingly.

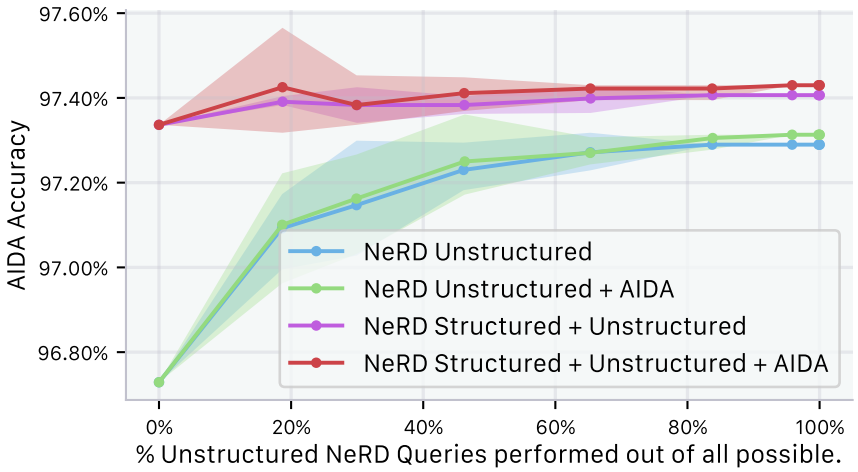
With prompt programming, it becomes possible to include external resources directly in the inputs of the model by using the idea of "open-book" [134] reasoning. The model's input can contain parenthetical and dynamic information regarding the user, time of day, or facts relevant to the main input that are impractical to memorize or may have changed since training. This capability has led to state of the art results in many Natural Language Processing (NLP) tasks thanks to zero or few-shot capabilities [19, 23, 85]. These works show that accuracy on few-shot tasks grows with the addition of prompting examples. However, including an entire training set as a prompt becomes impractical due to computation and memory constraints.

Retrieval-augmented models offer a solution by enabling a model to reference entire knowledge bases without growing the input. Using this approach, language models can attend to billions of tokens of unstructured data and reach equivalent performance with 15 to 25 times less parameters [14, 68]. The same trend is visible with retrieval over structured knowledge bases containing millions of entities and relations : DCA-SL+Triples[104] and DeepType 2 [124] greatly improve in Entity Linking accuracy by featurizing query results containing external facts and entity relations.

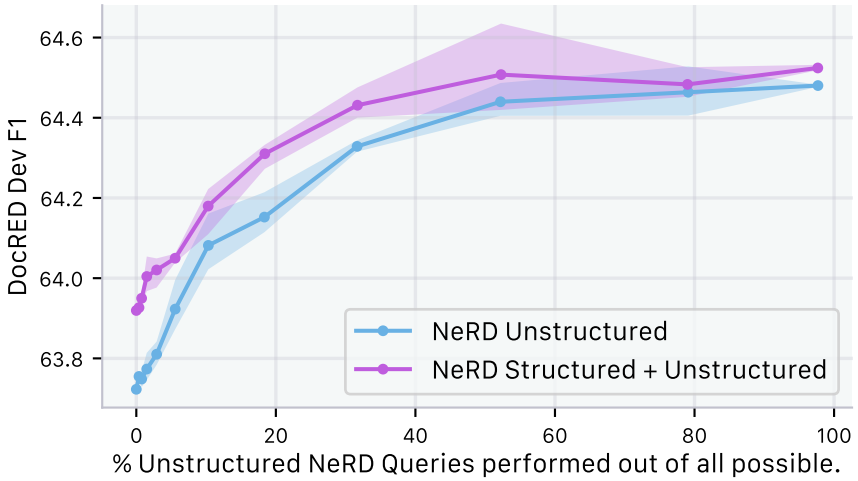
Unfortunately, retrieval-based approaches face a significant scaling challenge : while there is an abundance of unstructured data collectible on the web, we lack an equivalent way to automatically collect and curate structured data.

We overcome this limitation by using Pretrained Language Models to obtain a unified representation for structured and unstructured data. This lets us train retrieval models that can perform inference with either modality present. Our key contributions are twofold :

1. We present the Neural Relational Database (NeRD), a technique that enables retrieval models to learn how to perform inference using either a natural language representation of structured data, or unstructured data directly. We achieve this by requesting subject-relation-object triplets from structured and unstructured knowledge bases : structured results have a fixed set



(a) Entity Linking using DeepType 3



(b) Relation Extraction using RoBERTa-ATLOP + DeepType 3

FIGURE V.1 – DeepType 3's accuracy on AIDA and RoBERTa-ATLOP + DeepType 3's F1 on DocRED improves as we increase the size of the unstructured data knowledge base. Providing DeepType 3 with structured relations provides additional gains.

of potential relations (e.g. {"Maryam Mirzakhani", *occupation*, "Mathematician"}), while the unstructured results are made of text snippets found between the subject and object in a text corpus (e.g. {"Maryam Mirzakhani", "was awarded the", "Fields Medal"}). We unify the representation for these two modalities by embedding the triplet with the same frozen Pretrained Language Model and pooling all the results.

2. We propose DeepType 3, an Entity Linking system made by replacing the

relation retrieval from DeepType 2 with NeRD, and test its performance on Entity Linking and Relation Extraction and show improvements in both cases. We establish a new state of the art in Entity Linking performance when evaluating on seven commonly used datasets. Relation Extraction performance is measured on the DocRED dataset [180] where we obtain gains by combining existing Relation Extraction systems with DeepType 3. On both tasks, we demonstrate how NeRD's performance improves with additional structured or unstructured relations as visible in Figure V.1.

This chapter is structured as follows : Section 2 describes the NeRD technique and how it integrates in DeepType 3's neural network architecture. Section 3 presents our experiments comparing DeepType 3 to the Entity Linking and Relation Extraction state of the art and uses ablations to demonstrate the impact of structured and unstructured data in query results. In Section 4 we provide a discussion of the results and future work directions.

2 . Approach

2.1 . Unified Structured and Unstructured Relation Representation

Our proposed technique, NeRD, for unifying structured and unstructured relations relies on storing subject-relation-object triplets using natural language. We collect unstructured relations by finding text snippets between pairs of entities in a text corpus without any additional kind of pre or post-processing. Structured relations are converted to sound more natural and resemble unstructured relations (e.g. *employer* $\rightarrow$ "was employed by").

Results of either modality are turned when we query the knowledge base for relations between pairs of entities. Each triplet is converted into a sentence where the subject and object are replaced by a special "[BLANK]" token such as the mask token from BERT [32] before being embedded by a Pretrained Language Model Figure V.4. Because the natural language representation of a relation can be ambiguous or uninformative (See Figure V.5), we augment the relation representation by adding information about the type of subject and object. Specifically we obtain the neighboring Wikidata entities by following the "instance of" relation (e.g. human, country, etc.) and embed them. We concatenate the Pretrained Language Model activations with the "instance of" embeddings and project both of them to a common representation space using a fully connected layer as shown in Figure V.4.

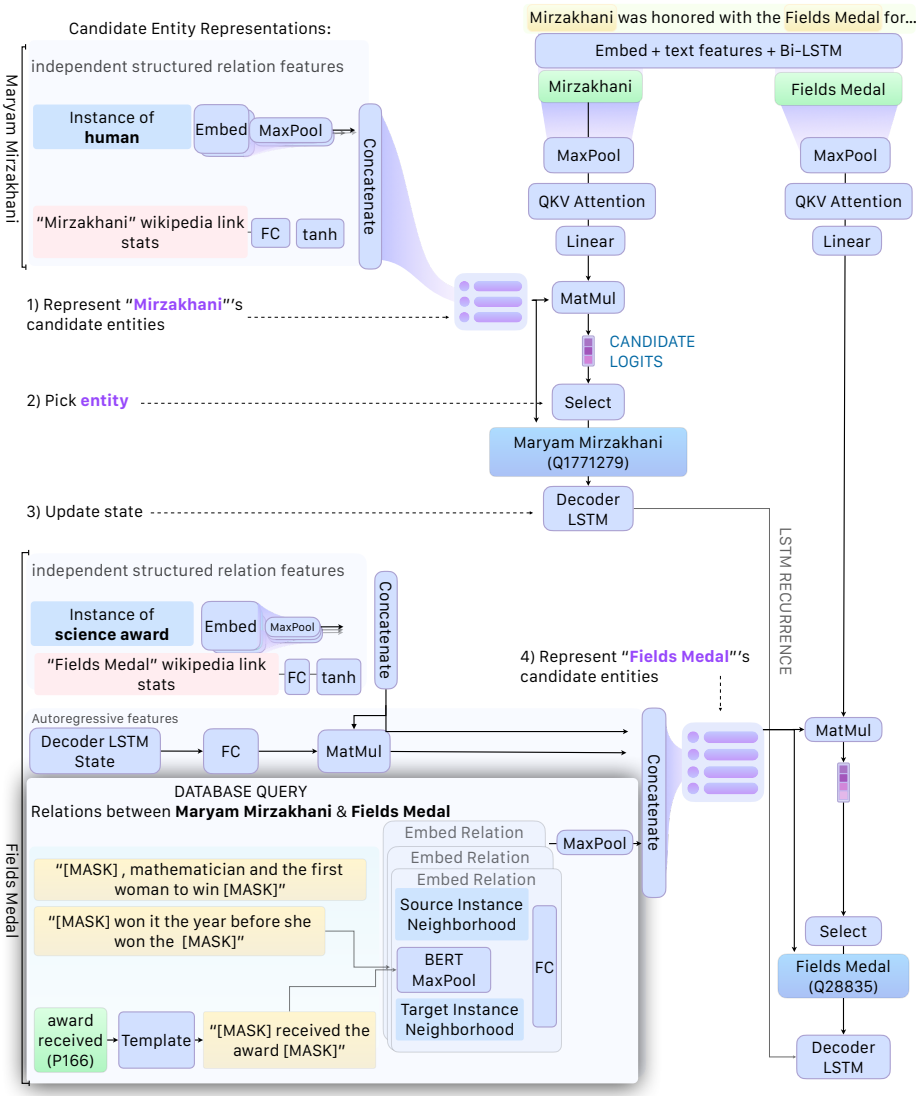


FIGURE V.2 – DeepType 3's neural network architecture showing how a document is read. DeepType 3's architecture builds upon the one used by DeepType 2, and adds the ability to obtain relational features by querying the Neural Relational Database as visible in the shadowed box at the bottom-left of the Figure. Starting from the top : an LSTM reads a document, while a separate graph neural network produces entity representations from the entity relations (1, 4). Each entity prediction (2) is fed to a Decoder LSTM (3) and added to the set of past predictions to perform future queries. Queries into the knowledge base seek relations between *entities predicted so far* and the *next mention's candidate entities*.

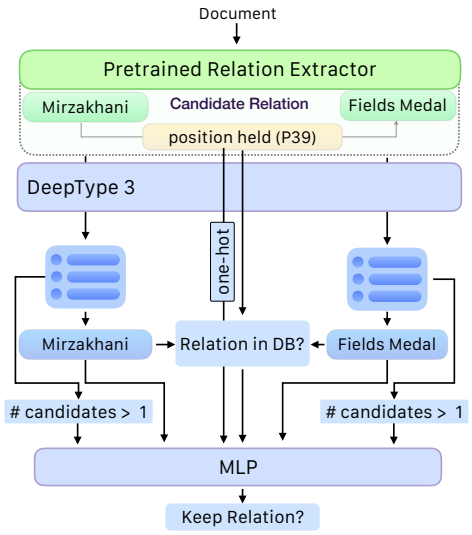


FIGURE V.3 – Combining DeepType 3 with a pretrained Relation Extractor to filter relations.

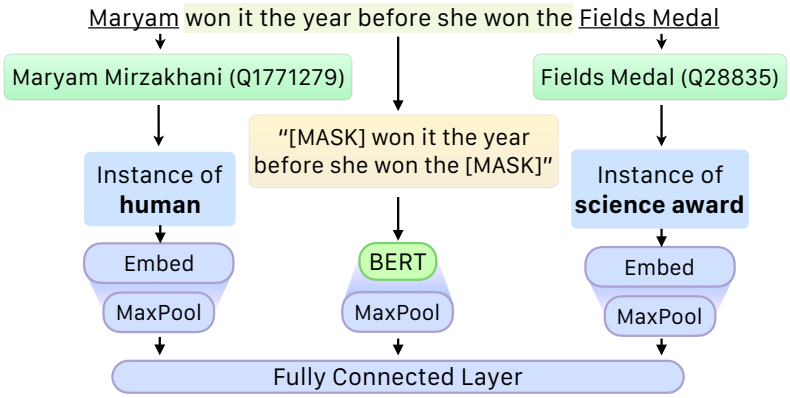


FIGURE V.4 – Structured and unstructured relations are represented using text. For unstructured relations we collect inter-entity text as a potential relation (1). To make relation representation more general we replace subject and target tokens by special mask tokens (2). The masked text outside its original context could lose crucial semantic information : we attempt to recover it with related entities from Wikidata about the subject and target (3). Variable-length representations for BERT and graph embeddings are max-pooled and linearly projected into a relation representation (4).

2.2 . DeepType 3's Neural Network Architecture

Overview We modify the current state of the art DeepType 2's architecture to accomodate features produced by NeRD when querying structured and unstructured relations. The mention and decoder LSTM components are unchanged.

Mention Representation DeepType 3's neural network architecture shown in Figure V.2 builds upon the one used in DeepType 2. As input the model receives a document D_m containing mentions. The document is tokenized, and each token is used to obtain a word, suffix, prefix embedding, and processed by a bidirectional-LSTM [51]. For mention in the document, a representation for mention is produced using a combination of max-pooling the Bi-LSTM hidden states for the relevant token positions, and using a QKV Attention operation [157] to augment the mention representation. The mention representation is then used as input to a dot product with candidate entity vectors (1). The result of this operation is a series of logits for each candidate entity (2).

After each mention, the state of the document is updated to reflect the newly selected entity (3). Newer candidate entities now also include contextual information such as the relation between newer candidates and past predictions.

In DeepType 3, the relation with past predictions is augmented to include unstructured relations obtained using the Neural Relational Database. At each prediction, a query is made into the database looking for matches for the candidate entity and a past prediction. The results of this query are unstructured textual relations shown in yellow boxes. The relation text is processed by a pretrained BERT model before being max-pooled and combined with the other candidate entity features to form the full candidate entity vector used to make a prediction in (4).

The bidirectional-LSTM activations corresponding to particular document-mention pair serve as an initial mention representation. To obtain a fixed size representation from the variable number of mention tokens, we apply a max-pooling procedure resulting in the mention vector : $h_{\mathrm{pool},m}(D_m)$.

Influence from longer range context can be induced on the mention representation by using the pooled vector $h_{\mathrm{pool},m}(D_m)$ as a key to a QKV Attention [157] operation over all document tokens. The result of this attention operation is the mention representation $h_{\mathrm{m}}(D_m)$ that is used to score candidate entities in Figure V.2).

We also associate to each mention a set of candidate entities using a lookup table, known as an *alias table* in the Entity Linking literature. The alias table maps the mention string to potential entities collected on a training corpora. Using an alias table enables a system to consider a small subset of entities from its KB when choosing which one might be present.

Candidate Entity Representation Entities are represented purely with relational features (Figure V.2.1, Figure V.2.4). We distinguish three sources for these features : 1) independent structured relations with neighbors, 2) relations with previously predicted entities returned by NeRD, 3) dot product with Decoder LSTM's hidden state.

Group Name (relations)	Vocab size	$C_{\min}$	d
Admin. territorial entity (P131)	17003	10	10
Instance/Subclass (P31, P279)	14624	5	40
Occupation (P106)	1421	10	10
Country (P27, P17, P495)	759	3	10
Sport/Industry (P101, P425, P1995, P641, P2578, P452)	599	10	40
Continent (P30)	12	10	10
Lat/Long (P625)	n/a	n/a	-
Birthdate (P569, P571, P585, P580, P577)	725	n/a	20
Gender (P21)	3	10	10

TABLE V.1 – Relations used for finding surrounding features and entities during entity representation. Embedding for the entities for has dimension d. $C_{\min}$ is the minimum occurence of a surrounding entity to be included in the embedding table.

1. Independent structured relations are obtained by finding all the entities surrounding the represented entity by following different Wikidata relations (e.g. instance of, nationality) detailed in Table V.1. Each relation produces a small set of entities that characterize the original entity as visible in Figure V.2a. We retain only the most frequently found related entities and embed those. We then max-pool each relation's group of entities, and concatenate the result.
2. The knowledge base is queried for relations between any previously predicted entity and the candidate entity e. The returned results use NeRD's relation representation to mask the differences between structured and unstructured relations. The results are max-pooled into a single vector summarizing the candidate entity e's relations with all previously predicted entities (Figure V.2.4). We subsample matches to only keep at most $N_{\text{train relations}}$ of them. This accelerates training and prevents the model from favoring one kind of relation modality over another.

3. We compute the dot product between the candidate entity e's representation and the hidden state of a Decoder LSTM (Figure V.2.3). After each prediction the Decoder LSTM updates its state using as input the selected entity's representation.

We concatenate these three feature groups into a candidate entity representation. We denote the history of past predictions by s, and the entity e's representation by $E(e, s, D_m)$.

Entity Prediction The mention representation D_m scores the candidate entities $c_0, \ldots, c_n$ returned by the alias table for this mention. The score is computed using the same approach as done in DeepType 2 (Subsubsection 2.1.5) using the dot-product between the mention and the candidate entity representation.

We jointly disambiguate document mentions D_m by selecting in left to right order the highest likelihood candidate. We either greedily pick the highest scoring entity or use beam search to approximately maximize the sequence of predictions (Figure V.2.2, Figure V.2.5). We treat the score of an entity c_i as an un-normalized log-likelihood among candidate entities $c_0, \ldots, c_n$ from the *alias table* :

$$\mathbb{P}(c_i|D_m, s) = \frac{\exp\left(h_{\mathrm{m}}(D_m) \cdot E(c_i, s, D_m)\right)}{\sum_{j=0}^{n} \exp\left(h_{\mathrm{m}}(D_m) \cdot E(c_j, s, D_m)\right)}. \tag{1}$$

Objective function The DeepType 3 Entity Linking (EL) system model parameters θ_{EL} are learnt using the same objective function as used in DeepType 2 defined in Subsubsection 2.1.6. The objective is to minimize $mathcal L_{\mathrm{EL}}(\theta_{\mathrm{EL}})$, the negative log likelihood of the ground truth entity e relative to the other candidates :

$$\mathcal{L}_{\mathrm{EL}}(\theta_{\mathrm{EL}}) = \sum_{\{e, D_m, s\}} -\log \mathbb{P}(e|D_m.s; \theta). \tag{2}$$

Certain examples have only the correct option available in the *alias table*, so no gradient would be provided by $\mathcal{L}_{\mathrm{EL}}(\theta_{\mathrm{EL}})$. We are able to avoid this issue by reusing the negative sample technique from DeepType 2 [124] described in Subsection 2.2 : we sample additional entities uniformly from our KB and use them as negative samples.

2.3 . Relation Extraction

Augmenting Relation Extractors With Entity Linking We use the Entity Linking system DeepType 3 to filter the predictions from a pretrained relation extraction model. We experiment with two relation extractors : RoBERTa-ATLOP [183] and DocuNet-RoBERTa [182]. We start by increasing the recall of the pretrained Relation Extractors at the expense of precision by lowering the prediction threshold. With this change the relation extraction now outputs "candidate relations" illustrated in Figure V.3.

We train a binary classifier $f_{\mathrm{RE}}(\cdot)$ to predict which of these candidate relations is correct (Figure V.3.2). The classifier is a multi-layer perceptron with ReLU activations and a sigmoid output activation. This classifier receives as input multiple features for each candidate relation :

— DeepType 3's entity representation for the entities detected in a mention pair,

— Boolean indicating whether the *alias table* had more than one match for either mention,

— Boolean indicating whether the predicted relation is present in Wikidata for : 1) this entity pair, 2) any pair of entities chosen among the mention candidates,

— One-hot vector for the predicted relation identifier.

The classifier filters the candidate relations.

Objective function In the Relation Extraction instantiation, we first train DeepType 3 to perform Entity Linking by minimizing $\mathcal{L}_{\mathrm{EL}}(\theta_{\mathrm{EL}})$ on an Entity Linking corpus. Next we use a pretrained Relation Extraction model to generate candidate relations on an Relation Extraction training corpus.

A binary classifier for Relation Extraction (RE) filtering $f_{\mathrm{RE}}(\cdot)$ with parameters θ_{RE} is trained to minimize the negative log likelihood $\mathcal{L}_{\mathrm{EL}}(\theta_{\mathrm{RE}})$ of the ground truth labels $Y_r = \{y_0, \ldots, y_n\}$ for each proposed relation $X_r = \{r_0, \ldots, r_n\}$ given the entities predicted by DeepType 3 :

$$\mathcal{L}_{\mathrm{RE}}(\theta_{\mathrm{RE}}) = \sum_{i=0}^{n} -\log \mathbb{P}(y_i|r_i), \tag{3}$$

$$= \sum_{i=0}^{n} -\log(f_{\mathrm{RE}}(r_i)) \cdot y_i - \log(1 - f_{\mathrm{RE}}(r_i)) \cdot (1 - y_i). \tag{4}$$

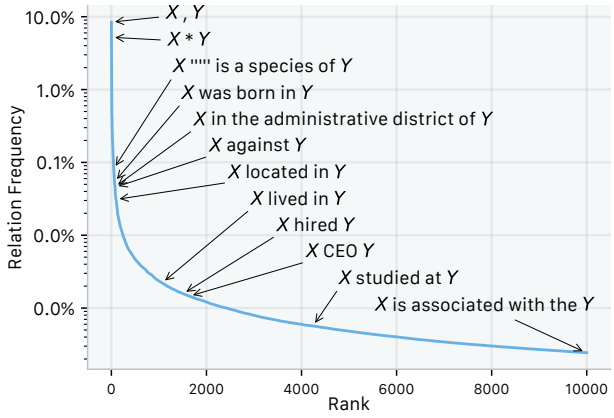


FIGURE V.5 – Frequency for the 10,000 most common text relations collected on the densified Wikipedia. Most are list-like ("*" is Wikipedia list markup). Noteworthy are employment, co-location, or competition (e.g. "against") relations.

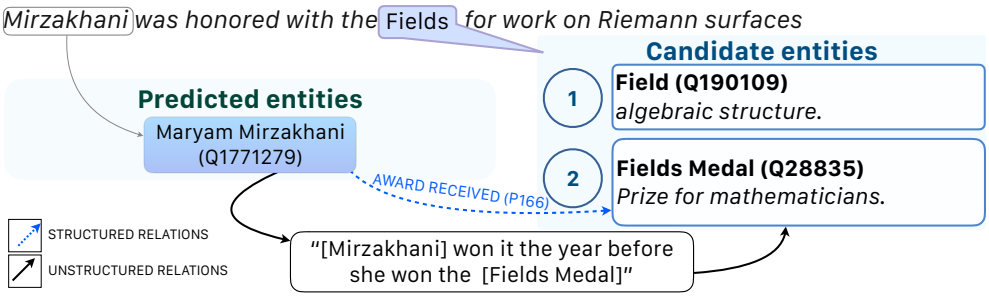


FIGURE V.6 – Structured and unstructured relations between "Fields Medal" and past predictions provide clues : candidate 2 has been referenced in the *award received* structured relation as well as in unstructured relations.

The knowledge base used by DeepType 3 contains structured and unstructured relations. Structured relations in this work come from Wikidata, but any other relational data between entities could also be used. The unstructured relations are obtained in an unsupervised fashion by taking all inter-entity strings from any sentence in external corpora such as Wikipedia. This collection procedure is high recall, containing both semantically meaningful relations such as "located in" or "hired" as visible in Figure V.5, but also more run-on sentences or compound statements of questionable utility. We empirically observe a difference in the content of these relations : structured relations describe a directed relation between subject and object (e.g. {"A", *parent*, "B"}), while unstructured relations are both directed

or reference a common third party (e.g. “is born in the same country as"). Identifying commonalities is particularly predictive of document coherency, and might help explain why structured and unstructured relations can be complementary. We illustrate the structured and unstructured relations in Figure V.6.

Source	Structured ?	Count	Unique
NeRD's Wikidata relations	✓	12M	21
Wikipedia	✗	37M	14M
Wikipedia Densified	✗	96M	36M
AIDA (train split)	✗	30,834	22,637

TABLE V.2 – Number of relations for each corpora.

Using Wikidata we obtain 12 million structured relations. These relations stem from 21 unique relation types such as occupation or employer, see Table V.3 for the full list. Next the strings containing between intra-Wikipedia links provide us with 37 million unstructured relations. We increase the number of extracted relations to 96 million as shown in Table V.2 by using the link densification strategy proposed in DeepType 2[124] described in Subsection 2.3.

3 . Results

3.1 . Experimental Setup

We train DeepType 3 to perform Entity Linking and Relation Extraction. We first evaluate the trained system on standard datasets. Second, we ablate the data stored in the knowledge base. Third, we investigate the effect of introducing novel unstructured data into the knowledge base.

In all our Entity Linking experiments DeepType 3 is trained for 1 million gradient steps on intra-wiki links using the same training corpus as DeepType 2 : the December 2017 English Wikipedia dump and AIDA's train split. For AIDA we use the PPR4NED *alias table* from [114] to facilitate comparisons with prior work. On other datasets, the *alias table* is constructed from the intra-wiki links on the English Wikipedia.

Hyperparameter name	Value
Input Bi-LSTM size	512
Input Bi-LSTM layers	2
Attention Heads	2
Attention Query size	128
Word embedding size	200
Word UNK Probability	0.25
Word Vocab Size	750,000
Word {Prefix,Suffix}-{2,3} embedding size	6
Word {Prefix,Suffix}-{2,3} vocab size	100,000
Wikipedia link stats Layer size	20
Wikipedia link stats Dropout probability	0.3
Wikipedia link stats power	0.18
Decoder LSTM size	128
Learning Rate	0.001
LR Decay/33,000 gradient steps	1%
LR Decay/400,000 gradient steps	80%
AIDA Train data oversampling	10
Negative Samples	20
Training max candidate entities	20
$N_{\text{train relations}}$	128

TABLE V.4 – Neural Network Hyperameters

Each experiment is conducted using two NVIDIA GTX 1080Ti on a computer with 128GB of RAM and 28 core 3.3Ghz Intel i9 CPUs. Training the model to completion takes about 9 days. The pretrained language model used to embed relations is a 104-language 768-dimensional, 12-layer, 12-head BERT [32]. DeepType 3's neural network dimensions and learning rate schedule are provided in Table V.4. The binary classifier $f_{\text{RE}}(\cdot)$ used to filter relations has two 100-dimensional hidden layers. Optimization uses the Adam optimizer with a learning rate decay schedule empirically determined using a Wikipedia-based validation set. Training batches are resized to contain at most 12,800 tokens per batch. If an out of memory error occurs, another batch is sampled instead, and training continues.

Model	Dev	
	F1	$F1_{ign}$
BERT ATLOP [183]	61.09	59.22
BERT ATLOP (ours)	61.18	59.27
+ DeepType 3	62.94	61.42
DeepType 3 Δ	+1.76	+2.15
RoBERTa-ATLOP [183]	63.18	61.32
RoBERTa-ATLOP (ours)	63.28±0.21	61.41±0.23
+ DeepType 3	64.53±0.05	63.04±0.06
DeepType 3 Δ	+1.25	+1.63
DocuNet-RoBERTa [182]	64.12	62.23
DocuNet-RoBERTa (ours)	63.67±0.05	61.73±0.06
+ DeepType 3	**65.71**±0.07	**64.11**±0.08
DeepType 3 Δ	+1.95	+2.38
SSAN-RoBERTa [174]	62.08	60.25
SSAN-RoBERTa + Adapt. [174]	65.69	63.76

TABLE V.5 – Relation Extraction results on DocRED Development dataset ($\mu \pm \sigma$, $N = 2$). Note : *SSAN-RoBERTa + Adapt.* uses additional data. Best results shown bold.

Model	Test	
	F1	$F1_{ign}$
RoBERTa-ATLOP [183]	63.40	61.39
RoBERTa-ATLOP (ours)	62.96	60.92
+ DeepType 3	63.56	61.47
DeepType 3 Δ	+0.60	+0.55
DocuNet-RoBERTa [182]	**64.55**	**62.39**
DocuNet-RoBERTa (ours)	63.52	61.36
+ DeepType 3	64.31	62.37
DeepType 3 Δ	+0.79	+1.01
SSAN-RoBERTa [174]	61.42	59.47
SSAN-RoBERTa + Adapt. [174]	65.92	63.78

TABLE V.6 – Relation Extraction results on DocRED Test dataset . Note : *SSAN-RoBERTa + Adapt.* uses additional data. Best results shown bold.

We apply DeepType 3 to relation extraction on the DocRED [180] dataset. Specifically we investigate two setups where DeepType 3 performs false positive relation rejection on the results of BERT-E/RoBERTa + ATLOP from [183], and DocuNet-RoBERTa [182]. We first reproduce the author's results and report their development set F1 scores in Table V.5. We then combine DeepType 3 with these trained models and observe gains in both setups. We submit our test set predictions to the online leaderboard, and show improvements on the Dev and Test sets for both BERT-E/RoBERTa + ATLOP and DocuNet-RoBERTa when they are combined with DeepType 3 in Table V.6.

3.3 . Entity Linking Evaluation

Model		TAC	AIDA
DeepType 3	μ	**97.74**	**97.87**
	σ	$\pm$0.14	$\pm$0.02
Human Oracle		96.86	96.78
DeepType 2		97.48	97.72
DeepType		90.9	94.9
Yang et al. [179]		-	95.9
De Cao et al. [29]		-	93.3
Févry et al. [42]		94.9	96.7-

TABLE V.7 – State of the art Entity Linking system accuracy on TAC and AIDA ($\mu \pm \sigma$, $N = 3$). Best results shown in bold.

Model		CWEB	WIKI	MSNBC	AQ.	ACE
DeepType 3	μ	**85.80**	**88.43**	**95.21**	**93.00**	**93.86**
	σ	$\pm$0.21	$\pm$0.55	$\pm$0.15	$\pm$0.94	$\pm$0.94
DeepType 2		85.57	87.83	95.12	92.74	92.23
Yang et al. [179]		81.8	79.2	92.6	89.9	89.2
De Cao et al. [29]		77.3	87.4	94.3	89.9	90.1

TABLE V.8 – State of the art Entity Linking system accuracy on other datasets ($\mu \pm \sigma$, $N = 3$). Best results shown in bold.

We evaluate DeepType 3 on seven standard entity linking datasets and compare to the human performance, DeepType and DeepType 2 and the current state of the art on TAC and AIDA in Table V.7, and other standard datasets in Table V.8. We report the mean and standard deviation across 3 training runs. Specifically we study the performance on the Wikipedia-based dataset WNED-WIKI [54], and news corpora datasets TAC-KBP 2010 [69], CoNLL (YAGO) AIDA [64] (test-b split), WNED-CWEB [54], MSNBC [27], AQUAINT (AQ.) [101], and ACE 2004 (ACE) [130]. Note that TAC-KBP 2010 documents only contain a single mention, so in order to use inter-entity relations we create additional mentions when phrases are found in our alias table. DeepType 3 attains superhuman performance on TAC and AIDA and we observe an improvement over the state of the art on all datasets.

3.4 . Memory Ablation

Memory	TAC	AIDA	DocRED Dev	
			F1	F1 IGN
No relations	94.62	96.73	63.73	62.18
Unstructured	96.42	97.28	64.48	62.95
Structured	95.75	97.34	63.92	62.38
Structured + Unstructured	**97.74**	**97.87**	**64.53**	**63.04**

TABLE V.9 – Performance impact of changing the relations stored in the knowledge base. Best results shown in bold.

On TAC, AIDA, and DocRED removing all relations from the knowledge base degrades performance. For the three data sets adding either Unstructured or Structured relations is a major improvement. TAC and DocRED profit most from unstructured relations and AIDA from structured relations. Adding both Structured and Unstructured relations provides the biggest boost as visible in Table V.9.

We now seek to validate whether our proposed technique to unify the representation of structured and unstructured data accomplishes the goal of making retrieval-models operate with either modality. We test this by toggling access to structured relations and varying the amount of structured relations returned by each query. As the unstructured relations return may change during subsampling, we report the mean performance and the min and max across 3 runs. In Figure V.1a we see Entity Linking performance improves with additional unstructured relations. A similar behavior is visible in Figure V.1b when the same experiment is run on relation extraction. The strongest performance in either task is reached when both relations kinds are present.

3.5 . Novel Relations

To truly test the ability for NeRD to enable extending structured databases with unseen unstructured data, we form new unstructured relations by collecting inter-entity text on AIDA's train split. Without retraining, we add these relations to the knowledge base and measure the performance in Figure V.1a. We find that DeepType 3 is able to leverage unseen relations to further improve its accuracy.

4 . Discussion

4.1 . Overview

NeRD creates a unified representation for structured and unstructured relations. This enables retrieval models to rely on either modality to solve a task. We achieve this by first converting all relations to a unified text representation, and second processing them by a Pretrained Language Model. To evaluate this technique we train DeepType 3, an Entity Linking system that performs retrieval over structured and unstructured data, and measure its performance on two tasks : Entity Linking and Relation Extraction. Capitalizing on over 108 million relations, this system improves the Relation Extraction performance of two different architectures and reaches a new state of the art in Entity Linking.

Ablations show how Entity Linking and Relation Extraction performance improves when additional structured or unstructured relations are added to the knowledge base. Peak performance is reached by combining structured and unstructured relations. We also investigate whether DeepType 3 can zero-shot use additional unstructured data : we add unstructure relations collected on AIDA's training split and observe that these lead to a slight performance gain.

4.2 . Future Work

The scope of our results is limited to commonly used Entity Linking datasets and the largest document-level Relation Extraction dataset. A useful extension is to verify whether these results hold in other domains where a rich mix of structured and unstructured relations is also present. A promising future work direction is to study how to continue scaling up the knowledge base by integrating non-relational unstructured data, and detect whether there is an emergent organization to the unstructured relations.

4.3 . Relation to Thesis

In this chapter we provided a solution to one of the Neuro-Symbolic challenges that motivates this thesis : enable an Artificial Intelligence system to self learn how to organize and represent external knowledge without human intervention. Moreover, experiments with the proposed approach indicate that structured and unstructured knowledge sources are complimentary, and the combined system still reaches superhuman accuracy. While the experiments focus on a only two downstream tasks, we can envision future Neuro-Symbolic systems that share the same capability to remain up to date, and organize information in a task-driven way in neural databases without human input.

Relation Type	Entity relation	Template
Identity	same entity	is the same as/is the same instance as
League	P118	is from the same league as/is in the
Admin Territorial Entity	P131	is located in/is from the same location as
Educated at	P69	was educated at
Political Party	P102	is a member of the political party/is from the same political party as
Spouse	P26	is married to
Country	P17	is from the country/is from the same country as
Sibling	P3373	is the sibling of
Employer	P108	worked at
Member of sports team	P54	is from the same team as/is a team member of
Sport	P641	plays the same sport as
US State	P131 (+ Q35657 is a parent)	is in the state of/is in the same state as

TABLE V.3 – Wikidata relations and their associated string templates when converting to a unified text representation. Note that certain templates exist either in a direct form (e.g. A is related to B through relation R) or indirect form (e.g. A and B both relate to C through R).

Discussion and Conclusion

ARTIFICIAL Intelligence is impacting the world in many transformative ways from industrial applications, to jobs requiring reasoning or even creativity. Present systems now exceed human performance in a wide variety of perception tasks, from speech recognition to object recognition, as well as challenging strategy games such as Go, Chess, or even Dota 2 [109].

However, progress in areas combining natural language understanding and fine grained knowledge about the world have been held back by the reliance of these systems on knowledge that grows stale, and the human effort required to ingest the vast amount of human knowledge available in symbolic form. Neuro-Symbolic systems have emerged as a solution to this challenge.

Advances in Neuro-Symbolic systems have come from two different research directions : 1) teaching a machine to control a symbolic structure or access external symbols, 2) modeling and representational improvements such as new actions, architectures, abstractions, or interfaces.

In this thesis, the two directions have been considered, yielding four contributions, respectively described in Chapter II, Chapter III, Chapter IV, and Chapter V.

In Chapter II, we introduce a human benchmark for Entity Linking. To the limits of our understanding we are the first to construct such as benchmark. The benchmark presented several difficulties because human performance can be underestimated when relying on non-expert crowd-sourced annotators. Specifically annotators can lack context, be distracted, or lack background knowledge. We were able to overcome these difficulties by performing an extensive trial, measuring group performance, and supplying annotators with suggestions ordered by Wikipedia usage frequency. Through this benchmark we noticed that performance on the TAC-KBP 2010 [69] dataset is abnormally lower for Artificial Intelligence systems due to the presence of single mentions per document. This benchmark also researchers to now perform a thorough error analysis such as the one done in Subsection 3.2.

In Chapter III, we contribute a new representation, neural type systems, that is based on the hierarchy of concepts present in online ontologies. This new representation presents several advantages relative to prior work, namely : transparency, sample efficiency, and higher accuracy. The representation is used to construct the state of the art Entity Linking system DeepType [126].

This work also showed connections with named entity recognition, and multi lingual representations. Specifically this forms of large scale trainings learn interlingual pivots. Similar observations have been made lately regarding different trainings leading to similarly structured latent spaces in [103].

In Chapter IV, we build upon the results of DeepType and extend the work to enable direct optimization of disambiguation accuracy. This work eliminates the manual effort required in DeepType such as membership rules, tuning, and the design of a type system. Instead, we are able to use the neighborhood surrounding an entity as its representation and a contrastive loss to favor one neighborhood

over another, without having to predict types. This enables richer types to exist, as they no longer have to be predicted.

Furthermore, this work introduce type interactions - abstract features that are typed, and describe relations between candidate entities and past predictions. These features highlight similarities such as shared employers, nationalities, or teams. We observe that this feature set is responsible for the majority of the gains.

Thanks to these new modeling improvements, we are better able to integrate external symbols without human effort. The added accuracy makes DeepType 2 [124] the first system to reach superhuman accuracy, outperforming the performance recorded in our human benchmark from Chapter III.

In Chapter V, we eliminate the reliance on structured knowledge bases by enabling a system to learn to become indifferent to stuctured and unstructured relations while retaining most of the performance of DeepType 2. In order to achieve this, we build upon DeepType 2's architecture and representation. Type interactions using typed relations are replaced by a unified text representation of structured and unstructured relations. The textual representation is processed by pre-trained Language Models providing a stable source of fixed length distributed representations encoding seen and unseen relations in a similar space. The neural network has no explicit signal indicating the source of the relations, and hence learns to interpret both sources of relations indifferently in the reasoning process.

We provide multiple ablations and experiments scaling the amount of relation data available in the database. We see improvements in performance with additional relation data, and gains from combining structured and unstructured relations.

The proposed approach is the new state of the art on standard entity linking datasets. As this systems receives a rich set of relation information, we experiment with combining DeepType 3 with pre-trained relation extraction systems. DeepType 3 is then trained to accept or reject proposed relations from the pre-trained relation extractor. The combined system outperforms the baseline, demonstrating gains from enabling access to entity and relation information when performing other downstream tasks.

Throughout the thesis we have proposed several directions of further research, such as representational extensions, additional languages and application domains, or scaling experiments in Chapters 2-5.

Future Perspectives

Several lessons can be taken from the work presented in this thesis. From the perspective of Artificial Intelligence progress towards human-level performance, we observe that approaches that are too reliant on human data are too fallible to omissions and mistakes, and systems that enforce exclusive reasoning using symbols often suffer from excessive rigidity and modeling errors. Neuro-Symbolic systems are able to make progress and even surpass human level performance when two conditions are met : 1) the training data has adequate coverage and diversity, 2) the neuro-symbolic reasoning process enables access to human knowledge with

sufficient fidelity, but retains sufficient flexibility to deviate when necessary.

Human Inspiration The first lesson concerns the importance of human benchmarks which push the field forward and inform us of new ways to design Artificial Intelligence systems. The works presented here showcase how abstracted models, coherency models based on relations, or even the design of a human benchmark surface strong inductive biases that unlock performance gains.

As future work in the short term, it would be valuable to extend reasoning benchmarks such as the Human Benchmark for Entity Linking presented into new settings and languages. The studied datasets TAC-KBP 2010 [69] and CoNLL AIDA (YAGO) [64] showcased difficulties with disambiguating journalistic shorthand and geographical ellipses (refering to a country by its capital city, or a sports team by the associated town) and we suspect that domains such as code-switching, multi-speaker, or with historical text will pose further difficulties by relying on context that is not always on the written page.

In the medium term, a useful extension would explore whether the Neuro-Symbolic framework can better approximate human inductive biases such as supporting more relaxed or dynamic forms of modeling of symbolic knowledge. Current practices use rigid decoding schemes, fixed neural network architectures, and static objective functions, whereas humans are effective as using a portfolio of different strategies to reason. Future work could explore whether a model can meta-learn better decoding algorithms, such as mixing Diffusion, Beam-Search, Mask Denoising, or changing the decoding order based on the inputs. Neural network architectures are generally fixed because deep neural network rely on layer inputs and outputs that have compatible shapes and are trained jointly. However, as we have presented in this work, a future neural network architecture could instead connect different sub-models using shared invariant representational spaces such as language, pixels, or even sound. Finally, a benefit of a changing objective function would be to emphasize current events, or specific reasoning patterns, or to reflect confidence in weakly supervised data. In the works presented we saw how enabling a model to treat type labels as inputs rather than supervision can induce better training by ignoring issues in the type label supervision. A future system could be iteratively retrained based on which patterns were effective in practice, or learn to discard data that is unreliable. Thanks to these developments, it would then become a lot easier for online-learning systems to take advantage of the Neural Relational Database's extensible nature while minimizing the risk that erroneous data poisons the rest of the model.

...being able to compress well is closely related to acting intelligently...

Marcus Hutter

Abstraction and Symbol Manipulation Another direction that is particularly important within Artificial Intelligence research is the ability to meta-learn through abstraction and symbol manipulation. Indeed, the impressive gains from prompt-programming of language models, or those observed from DeepType systems come from enabling systems to operate a higher level of abstraction than just the entities or words. In this work we show how allowing machines to observe and manipulate human hierarchies of concepts provides new reasoning capabilities that improve accuracy beyond what was possible before. Similar gains were observed by providing symbols to manipulate as scratch pads to language models [108], or forcing models to elucidate their reasoning by asking to think step by step [79].

In the short term future work involves simplifying certain modeling decisions to increase the meta-reasoning and abstraction of a task. All present Entity Linking systems make instance-level decisions for each mention, however many of these decisions are repetitive and obfuscate complex causal factors such as coreference, speaker order, or external context. Following the intuition of the Hutter Prize and the work that inspired it [67] we can seek to obtain a more compressed prediction format that focuses model capacity towards the more important aspects of the problem. For instance, future work could opt to make a first sequence of decisions seeking to discover which mentions refer to the same entity, and second, what that entity should be. By performing several rounds of these meta predictions, such as enforcing specific relations between entities each mention refers to, we constrain the search space and eliminate many spurious possibilities. Because the number of potential meta-predictions is limited, an entire corpus could be distilled into several general prediction patterns that a model has to capture. Each step in this direction, further increases the ability for a model to perform the bulk of its reasoning in an abstract way.

In the medium term, the ability to relabel datasets to support abstract reasoning is bottle-necked by the availability of relational datasets that can augment the original labels. In order to supervise abstract meta-reasoning in new domains such as computer vision or speech recognition we will need new sources of augmentation, such as teaching models to self-rewrite the label without access to the external relational datasets as augmentation. Future work could investigate whether the ability to zero and few-shot learn in large language models can be applied towards label meta-rewriting.

End-to-End Robustness We have also seen how end-to-end learning of a model enables higher accuracy by eliminating issues with errors or omissions in the knowledge base. While DeepType relies heavily on a human designed type system, and makes decisions using soft constraints based on existing concept hierarchies, DeepType 2 is able to outperform by being able to reweigh aspects of the types directly without the need for rules to organize the type system. DeepType 3 goes one step further, and complements knowledge bases such as Wikidata and Wikipedia with unstructured relations obtained from text corpora. This addition of new relations increases coverage, while also ensuring that the modeling of relations is continuous and able to generalize better between similar relations or adapt to unseen ones. Future Artificial Intelligence systems that scale and improve will require DeepType 2's robustness to errors and omissions in their data, and DeepType 3's ability to extend a knowledge base online.

In the short term an obvious extension of the DeepType systems involves increasing the scope of the prediction to include mention detection. The supervision and benchmark datasets in Entity Linking have historically broken the tasks into mention prediction and entity linking, however recent works such as [80] propose to construct systems that perform both tasks. Future work in this direction is meaningful because it greatly simplifies the Entity Linking systems, and eliminates a huge source of error and variability. However, end-to-end entity linking is also challenging because present evaluation methodologies will become incompatible with the outputs of models that may disagree on the document mentions and their spans. A further difficulty in developing end-to-end Entity Linking systems is the inability to constrain the solution space by considering the joint set of mentions and an alias table, rather constraints will have to be discovered by a model or made possible by allowing a model to query an external knowledge base for those explicitly. Finally, rare mentions and entities are currently greatly assisted by relying on explicit lookup tables such as alias tables, which do not have any of the recall and learning capacity issues of neural networks.

In the medium term, future work should instead focus on the downstream tasks where entity linking is currently a useful feature. Today the output of an Entity Linking system is a feature for information retrieval, machine translation, automated trading systems, or one component in a larger natural language understanding system. Switching the metrics in Entity Linking from accuracy on specialized datasets to improving the precision and recall of information retrieval systems will better align the incentives and ensure that progress is being made towards the true end goals. In practice this means that an extension could use an objective function where certain labels matter less, such as minor technical terms in a larger document in an information retrieval system, while others are emphasized, such as ensuring that central concepts such as characters in a story are perfectly detected when translating or summarizing a story.

AI is whatever hasn't been done yet.

Larry Tesler

Closing Remarks Artificial Intelligence is advancing at an unbelievable pace, overcoming challenges that were thought to be still years away from protein folding [73], to playing Go [147], or generating photorealistic images [140]. Each of these advances brings us closer to moonshot goals such as Artificial General Intelligence, but in the words of Larry Tesler "AI is whatever hasn't been done yet" the appearance of intelligence vanishes whenever a milestone is reached.

A glimmer of hope is nonetheless visible in a new class of Artificial Intelligence systems. These new Artificial Intelligence systems have capabilities that are not fully revealed upon their design and construction. Rather, we see Artificial Intelligence systems for strategic games that use moves and tactics that human experts had not even imagined, large language models that can imitate new patterns and tasks on the fly, and image generation models that can make associations and compositions unseen in their training data. These Artificial Intelligence systems demonstrate much broader skills than ever before.

These breakthroughs can be seen through the lens of two key Neuro-Symbolic attributes : a meta-learning inductive bias, and massive training corpora. The Meta-learning inductive bias is caused by forcing models to manipulate symbolic structures such as game boards, language scratch pads, or image canvases. While the massive amounts of training data act like an online learning setup which further reinforces the ability to meta-learn, and the data volume and diversity give the models robust priors about the world. Throughout this thesis we have studied the effects of these ideas on our own Artificial Intelligence systems and unlocked superhuman capabilities.

We leave the reader with the following open question : is scale all we need to unlock the next leap in Artificial Intelligence ?

ANNEXE A

DeepType Appendix

1 . Human Type System

Shown below are the 5 different type axes designed by humans.

Post-1950
Pre-1950
Other

TABLE A.1 – Human Type Axis : Time

Africa
Antarctica
Asia
Europe
Middle East
North America
Oceania
Outer Space
Populated place unlocalized
South America
Other

TABLE A.2 – Human Type Axis : Location

Activity	Genre	Radio program
Aircraft	Geographical object	Railroad
Airport	Geometric shape	Record chart
Algorithm	Hazard	Region
Alphabet	Human	Religion
Anatomical structure	Human female	Research
Astronomical object	Human male	River
Audio visual work	International relations	Road vehicle
Award	Kinship	Sea
Award ceremony	Lake	Sexual orientation
Battle	Language	Software
Book magazine article	Law	Song
Brand	Legal action	Speech
Bridge	Legal case	Sport
Character	Legislative term	Sport event
Chemical compound	Mathematical object	Sports terminology
Clothing	Mind	Strategy
Color	Molecule	Taxon
Concept	Monument	Taxonomic rank
Country	Mountain	Title
Crime	Musical work	Train station
Currency	Name	Union
Data format	Natural phenomenon	Unit of mass
Date	Number	Value
Developmental biology period	Organization	Vehicle
Disease	Other art work	Vehicle brand
Electromagnetic wave	People	Volcano
Event	Person role	War
Facility	Physical object	Watercraft
Family	Physical quantity	Weapon
Fictional character	Plant	Website
Food	Populated place	Other
Gas	Position	
Gene	Postal code	

TABLE A.3 – Human Type Axis : IsA

Archaeology	Health insurance	Science histology
Automotive industry	Health life insurance	Science meteorology
Aviation	Health medical	Sex industry
Biology	Health med activism	Smoking
Botany	Health med doctors	Sport-air-sport
Business other	Health med society	Sport-american football
Construction	Health organisations	Sport-athletics
Culture	Health people in health	Sport-australian football
Culture-comics	Health pharma	Sport-baseball
Culture-dance	Health protein	Sport-basketball
Culture-movie	Health protein wkp	Sport-climbing
Culture-music	Health science medicine	Sport-combat sport
Culture-painting	Heavy industry	Sport-cricket
Culture-photography	Home	Sport-cue sport
Culture-sculpture	Hortculture and gardening	Sport-cycling
Culture-theatre	Labour	Sport-darts
Culture arts other	Law	Sport-dog-sport
Culture ceramic art	Media	Sport-equestrian sport
Culture circus	Military war crime	Sport-field hockey
Culture literature	Nature	Sport-golf
Economics	Nature-ecology	Sport-handball
Education	Philosophy	Sport-ice hockey
Electronics	Politics	Sport-mind sport
Energy	Populated places	Sport-motor sport
Engineering	Religion	Sport-multisports
Environment	Retail other	Sport-other
Family	Science other	Sport-racquet sport
Fashion	Science-anthropology	Sport-rugby
Finance	Science-astronomy	Sport-shooting
Food	Science-biophysics	Sport-soccer
Health-alternative-medicine	Science-chemistry	Sport-strength-sport
Health-science-audiology	Science-computer science	Sport-swimming
Health-science-biotechnology	Science-geography	Sport-volleyball
Healthcare	Science-geology	Sport-winter sport
Health cell	Science-history	Sport water sport
Health childbrith	Science-mathematics	Toiletry
Health drug	Science-physics	Tourism
Health gene	Science-psychology	Transportation
Health hospital	Science-social science other	Other
Health human gene	Science chronology	

TABLE A.4 – Human Type Axis : Topic

Bibliographie

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m : A large-scale video classification benchmark. *arXiv preprint arXiv :1609.08675*, 2016.

[2] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv :1910.07113*, 2019.

[3] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016.

[4] D. Amodei, D. Hernandez, G. Sastry, J. Clark, G. Brockman, and I. Sutskever. Ai and compute, 2018. URL Heruntergeladenvonhttps://blog.openai.com/aiand-compute.

[5] C. Anderson. The end of theory : The data deluge makes the scientific method obsolete, 2008. URL https://www.wired.com/2008/06/pb-theory/.

[6] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins. Globally normalized transition-based neural networks. *arXiv preprint arXiv :1603.06042*, 2016.

[7] S. Ö. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, et al. Deep voice : Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017.

[8] S. Bader and P. Hitzler. Dimensions of neural-symbolic integration-a structured survey. *arXiv preprint cs/0511042*, 2005.

[9] B. Bandyopadhyay, P. Maneriker, V. Patel, S. Y. Sahai, P. Zhang, and S. Parthasarathy. DrugDBEmbed : Semantic queries on relational database using supervised column encodings. *arXiv preprint arXiv :2007.02384*, 2020.

[10] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79 (1) :151–175, 2010.

[11] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis. Machine learning in agriculture : A comprehensive updated review. *Sensors*, 21(11) :3758, 2021.

[12] S. K. Biswal and N. K. Gouda. Artificial intelligence in journalism : A boon or bane ? In *Optimization in machine learning and applications*, pages 155–167. Springer, 2020.

[13] R. Bordawekar, B. Bandyopadhyay, and O. Shmueli. Cognitive database : A step towards endowing relational databases with artificial intelligence capabilities. *arXiv preprint arXiv :1712.07199*, 2017.

[14] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pages 2206–2240. PMLR, 2022.

[15] R. Botha and C. Knight. *The cradle of language*, volume 12. OUP Oxford, 2009.

[16] V. A. Brei et al. Machine learning in marketing : Overview, learning strategies, applications, and future developments. *Foundations and Trends® in Marketing*, 14(3) :173–236, 2020.

[17] S. Bringsjord, P. Bello, and D. Ferrucci. Creativity, the turing test, and the (better) lovelace test. In *The Turing Test*, pages 215–239. Springer, 2003.

[18] S. Broscheit, K. Gashteovski, Y. Wang, and R. Gemulla. Can we predict new facts with open knowledge graph embeddings ? a benchmark for open link prediction. In *ACL 2020 : the 58th Annual Meeting of the Association for Computational Linguistics, proceedings of the conference*. Association for Computational Linguistics, 2020.

[19] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, 2020.

[20] J. Chen, M. Hu, B. Li, and M. Elhoseiny. Efficient self-supervised vision pretraining with local masked reconstruction. *arXiv preprint arXiv :2206.00790*, 2022.

[21] J. P. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv :1511.08308*, 2015.

[22] N. Chomsky. *Powers and prospects : Reflections on nature and the social order*. Haymarket Books+ ORM, 2015.

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

[24] A. D. Cohen, S. Rosenman, and Y. Goldberg. Relation classification as two-way span-prediction. *arXiv preprint arXiv :2010.04829*, 2020.

[25] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE) :2493–2537, 2011.

[26] L. Columbus. 10 ways ai has the potential to improve agriculture in 2021, 2021. URL https://www.forbes.com/sites/louiscolumbus/2021/02/17/10-ways-ai-has-the-potential-to-improve-agriculture-in-2021/?sh=5d7d82157f3b.

[27] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 708–716, 2007.

[28] H. T. Dang, K. Owczarzak, et al. Overview of the tac 2008 update summarization task. In *TAC*, 2008.

[29] N. De Cao, G. Izacard, S. Riedel, and F. Petroni. Autoregressive entity retrieval. *arXiv preprint arXiv :2010.00904*, 2020.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet : A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[31] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets : Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3450–3457. IEEE, 2012.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert : Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv :1810.04805*, 2018.

[33] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox : A generative model for music. *arXiv preprint arXiv :2005.00341*, 2020.

[34] A. Dogan and D. Birant. Machine learning and data mining in manufacturing. *Expert Systems with Applications*, 166 :114060, 2021.

[35] N. Drummond and R. Shearer. The open world assumption. In *eSI Workshop : The Closed World of Databases meets the Open World of the Semantic Web*, volume 15, page 1, 2006.

[36] L. Du, A. Kumar, M. Johnson, and M. Ciaramita. Using entity information from a knowledge base to improve relation extraction. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 31–38, 2015.

[37] K. Ellis, M. Nye, Y. Pu, F. Sosa, J. Tenenbaum, and A. Solar-Lezama. Write, execute, assess : Program synthesis with a REPL. In *Advances in Neural Information Processing Systems*, pages 9169–9178, 2019.

[38] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy. Named entity disambiguation for noisy text. *arXiv preprint arXiv :1706.09147*, 2017.

[39] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam. Open information extraction : The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[40] European Commission. Proposal for a regulation laying down harmonised rules on artificial intelligence (artificial intelligence act), 2021. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206.

[41] P. Ferragina and U. Scaiella. Tagme : on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.

[42] T. Févry, N. FitzGerald, L. B. Soares, and T. Kwiatkowski. Empirical evaluation of pretraining strategies for supervised entity linking. *arXiv preprint arXiv :2005.14253*, 2020.

[43] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5) :378, 1971.

[44] M. Fortunato, M. Tan, R. Faulkner, S. Hansen, A. P. Badia, G. Buttimore, C. Deck, J. Z. Leibo, and C. Blundell. Generalization of reinforcement learners with working and episodic memory. *arXiv preprint arXiv :1910.13406*, 2019.

[45] L. Galárraga, G. Heitz, K. Murphy, and F. M. Suchanek. Canonicalizing open knowledge bases. In *Proceedings of the 23rd acm international conference on conference on information and knowledge management*, pages 1679–1688, 2014.

[46] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler. Get3d : A generative model of high quality 3d textured shapes learned from images. *arXiv preprint arXiv :2209.11163*, 2022.

[47] A. d. Garcez and L. C. Lamb. Neurosymbolic ai : the 3rd wave. *arXiv preprint arXiv :2012.05876*, 2020.

[48] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou. Deep voice 2 : Multi-speaker neural text-to-speech. *Advances in neural information processing systems*, 30, 2017.

[49] A. Globerson, N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringaard, and F. Pereira. Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, volume 1, pages 621–631, 2016.

[50] K. Grace, J. Salvatier, A. Dafoe, B. Zhang, and O. Evans. When will ai exceed human performance ? evidence from ai experts. *Journal of Artificial Intelligence Research*, 62 :729–754, 2018.

[51] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18(5-6) :602–610, 2005.

[52] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv :1410.5401*, 2014.

[53] T. Gruetze, G. Kasneci, Z. Zuo, and F. Naumann. Coheel : Coherent and efficient named entity linking through random walks. *Journal of Web Semantics*, 37 :75–89, 2016.

[54] Z. Guo and D. Barbosa. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4) :459–479, 2018.

[55] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation : A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[56] M. Harrison. Ai that generates music from prompts should probably scare musicians, 2022. URL https://futurism.com/the-byte/ai-music-text-prompts.

[57] I. Harvey. The microbial genetic algorithm. In *European Conference on Artificial Life*, pages 126–133. Springer, 2009.

[58] F. Hasibi, K. Balog, and S. E. Bratsberg. On the reproducibility of the tagme entity linking system. In *European Conference on Information Retrieval*, pages 436–449. Springer, 2016.

[59] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[60] P. He, X. Liu, J. Gao, and W. Chen. Microsoft deberta surpasses human performance on the superglue benchmark. *Microsoft, Redmond. Accessed : Nov*, 18, 2021.

[61] G. E. Hinton et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

[62] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, et al. Imagen video : High definition video generation with diffusion models. *arXiv preprint arXiv :2210.02303*, 2022.

[63] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.

[64] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, Scotland*, pages 782–792, 2011.

[65] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv :1801.06146*, 2018.

[66] F.-h. Hsu, M. S. Campbell, and A. J. Hoane Jr. Deep blue system overview. In *Proceedings of the 9th international conference on Supercomputing*, pages 240–244, 1995.

[67] M. Hutter. Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions. In *European conference on machine learning*, pages 226–238. Springer, 2001.

[68] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv :2208.03299*, 2022.

[69] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, volume 3, pages 3–3, 2010.

[70] H. Joko and F. Hasibi. Personal entity, concept, and named entity linking in conversations. *arXiv preprint arXiv :2206.07836*, 2022.

[71] M. I. Jordan. Serial order : A parallel distributed processing approach. Technical report, California University San Diego, La Jolla, Institute for Cognitive Science, 1986.

[72] A. Joulin and T. Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. *arXiv preprint arXiv :1503.01007*, 2015.

[73] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873) :583–589, 2021.

[74] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv :1705.06950*, 2017.

[75] H. J. Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10) : 947–954, 1960.

[76] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. *arXiv preprint arXiv :1508.06615*, 2015.

[77] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[78] D. P. Kingma and J. Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[79] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv :2205.11916*, 2022.

[80] N. Kolitsas, O.-E. Ganea, and T. Hofmann. End-to-end neural entity linking. *arXiv preprint arXiv :1808.07699*, 2018.

[81] J. Krishnamurthy, P. Dasigi, and M. Gardner. Neural semantic parsing with type constraints for semi-structured tables. In *EMNLP*, volume 17, pages 1532–1543, 2017.

[82] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pages 282–289, 2001.

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

[84] P. Le and I. Titov. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1595–1604, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi : 10.18653/v1/P18-1148. URL https://www.aclweb.org/anthology/P18-1148.

[85] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv :2206.14858*, 2022.

[86] F. Li, M. Zhang, G. Fu, and D. Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1) :1–11, 2017.

[87] J. Ling, N. FitzGerald, Z. Shan, L. B. Soares, T. Févry, D. Weiss, and T. Kwiatkowski. Learning cross-context entity representations from text. *arXiv preprint arXiv :2001.03765*, 2020.

[88] X. Ling and D. S. Weld. Fine-grained entity recognition. In *AAAI*. Citeseer, 2012.

[89] X. Ling, S. Singh, and D. S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3 :315–328, 2015.

[90] R. Liu, J. Wei, S. S. Gu, T.-Y. Wu, S. Vosoughi, C. Cui, D. Zhou, and A. M. Dai. Mind's eye : Grounded language model reasoning through simulation. *arXiv preprint arXiv :2210.05359*, 2022.

[91] X. Liu, P. He, W. Chen, and J. Gao. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv :1904.09482*, 2019.

[92] R. Loews. Alphafold is the most important achievement in ai – ever, 2021. URL https://www.forbes.com/sites/robtoews/2021/10/03/

alphafold-is-the-most-important-achievement-in-ai-ever/?sh=5c8266c26e0a.

[93] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee. Zero-shot entity linking by reading entity descriptions. *arXiv preprint arXiv :1906.07348*, 2019.

[94] S. Magnolini, V. Piccioni, V. Balaraman, M. Guerini, and B. Magnini. How to use gazetteers for entity recognition with neural models. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 40–49, 2019.

[95] C. D. Manning. Part-of-speech tagging from 97% to 100% : is it time for some linguistics ? In *International conference on intelligent text processing and computational linguistics*, pages 171–189. Springer, 2011.

[96] G. V. Maverick. Computational analysis of present-day american english, 1969.

[97] B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation : Contextualized word vectors. *Advances in neural information processing systems*, 30, 2017.

[98] J. McCarthy. From here to human-level ai. *Artificial Intelligence*, 171(18) : 1174–1182, 2007.

[99] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[100] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf : Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1) :99–106, dec 2021. ISSN 0001-0782. doi : 10.1145/3503250. URL https://doi.org/10.1145/3503250.

[101] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[102] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862) :207–212, 2021.

[103] L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodolà. Relative representations enable zero-shot latent space communication. *arXiv preprint arXiv :2209.15430*, 2022.

[104] I. O. Mulang', K. Singh, C. Prabhu, A. Nadgeri, J. Hoffart, and J. Lehmann. Evaluating the impact of knowledge graph context on entity disambiguation models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2157–2160, 2020.

[105] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, et al. Webgpt : Browser-assisted question-answering with human feedback. *arXiv preprint arXiv :2112.09332*, 2021.

[106] F. Nie, Y. Cao, J. Wang, C.-Y. Lin, and R. Pan. Mention and entity description co-attention for entity disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[107] J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al. Universal dependencies v1 : A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

[108] M. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, et al. Show your work : Scratchpads for intermediate computation with language models. *arXiv preprint arXiv :2112.00114*, 2021.

[109] OpenAI, :, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning, 2019. URL https://arxiv.org/abs/1912.06680.

[110] D. Paperno, G. Kruszewski, A. Lazaridou, Q. N. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. Fernández. The lambada dataset : Word prediction requiring a broad discourse context. *arXiv preprint arXiv :1606.06031*, 2016.

[111] C. Parsing. Speech and language processing. 2009.

[112] J. Pennington, R. Socher, and C. D. Manning. Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[113] C. Perreault and S. Mathew. Dating the origin of language using phonemic diversity. *PloS one*, 7(4) :e35289, 2012.

[114] M. Pershina, Y. He, and R. Grishman. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 238–243, 2015.

[115] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. arxiv 2018. *arXiv preprint arXiv :1802.05365*, 12, 1802.

[116] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller. Deep voice 3 : Scaling text-to-speech with convolutional sequence learning. In *International Conference on Learning Representations*, 2018.

[117] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion : Text-to-3d using 2d diffusion. *arXiv preprint arXiv :2209.14988*, 2022.

[118] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang. Conll-2012 shared task : Modeling multilingual unrestricted coreference in ontonotes. In *EMNLP-CoNLL Shared Task*, 2012.

[119] A. Radford, R. Jozefowicz, and I. Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv :1704.01444*, 2017.

[120] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[121] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8) :9, 2019.

[122] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[123] 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. *arXiv preprint arXiv :1910.10683*, 2019.

[124] J. Raiman. Deeptype 2 : Superhuman entity linking all you need is type interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[125] J. Raiman and J. Miller. Globally normalized reader. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1059–1069, 2017.

[126] J. Raiman and O. Raiman. Deeptype : multilingual entity linking by neural type system evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[127] P. Ramachandran, P. J. Liu, and Q. V. Le. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv :1611.02683*, 2016.

[128] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv :2204.06125*, 2022.

[129] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

[130] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th annual meeting of the association for computational linguistics : Human language technologies*, pages 1375–1384, 2011.

[131] R. Reiter. *On Closed World Data Bases*, pages 55–76. Springer US, Boston, MA, 1978. ISBN 978-1-4684-3384-5. doi : 10.1007/978-1-4684-3384-5_3. URL https://doi.org/10.1007/978-1-4684-3384-5_3.

[132] S. Ritter, J. Wang, Z. Kurth-Nelson, S. Jayakumar, C. Blundell, R. Pascanu, and M. Botvinick. Been there, done that : Meta-learning with episodic recall. In *International Conference on Machine Learning*, pages 4354–4363. PMLR, 2018.

[133] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[134] A. Roberts, C. Raffel, and N. Shazeer. How much knowledge can you pack into the parameters of a language model ? *arXiv preprint arXiv :2002.08910*, 2020.

[135] H. Rosales-Méndez, A. Hogan, and B. Poblete. Voxel : a benchmark dataset for multilingual entity linking. In *International Semantic Web Conference*, pages 170–186. Springer, 2018.

[136] F. Rossi. Thinking fast and slow in ai (aaai 2022 invited talk), 2022. URL https://aaai-2022.virtualchair.net/plenary_13.html.

[137] R. Roy, J. Raiman, N. Kant, I. Elkin, R. Kirby, M. Siu, S. Oberman, S. Godil, and B. Catanzaro. Prefixrl : Optimization of parallel prefix circuits using deep reinforcement learning. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 853–858. IEEE, 2021.

[138] R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2) : 127–190, 1999.

[139] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088) :533–536, 1986.

[140] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv :2205.11487*, 2022.

[141] E. F. T. K. Sang and F. D. Meulder. Introduction to the conll-2003 shared task : Language-independent named entity recognition. In *CoNLL*, 2003.

[142] C. Sang-Hun and J. Markoff. Master of go board game is walloped by google computer program. *The New York Times*. URL https://www.nytimes.com/2016/03/10/world/asia/google-alphago-lee-se-dol.html.

[143] J. C. Schlimmer and R. Granger. Beyond incremental processing : Tracking concept drift. In *AAAI*, 1986.

[144] B. Selman. Aaai2022 : Presidential address : The state of ai, 2022. URL https://aaai-2022.virtualchair.net/plenary_2.html.

[145] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3) :379–423, 1948.

[146] A. Sil, G. Kundu, R. Florian, and W. Hamza. Neural cross-lingual entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[147] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676) :354–359, 2017.

[148] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, et al. Make-a-video : Text-to-video generation without text-video data. *arXiv preprint arXiv :2209.14792*, 2022.

[149] C. S. Smith. A.i. here, there, everywhere, 2021. URL https://www.nytimes.com/2021/02/23/technology/ai-innovation-privacy-seniors-education.html.

[150] N. Spasojevic, P. Bhargava, and G. Hu. Dawt : Densely annotated wikipedia texts across multiple languages. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1655–1662, 2017.

[151] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv :1505.00387*, 2015.

[152] N. Statt. Openai's dota 2 ai steamrolls world champion e-sports team with back-to-back victories. *The Verge*. URL https://www.theverge.com/2019/4/13/18309459/openai-five-dota-2-finals-ai-bot-competition-og-e-sports-the-international-champion.

[153] R. Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13 :12, 2019.

[154] A. Tizghadam, H. Khazaei, M. H. Moghaddam, and Y. Hassan. Machine learning in transportation, 2019.

[155] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[156] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(October) : 433–60, 1950. doi : 10.1093/mind/lix.236.433.

[157] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[158] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan. Phenaki : Variable length video generation from open domain textual description. *arXiv preprint arXiv :2210.02399*, 2022.

[159] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. Grammar as a foreign language. *Advances in neural information processing systems*, 28, 2015.

[160] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782) :350–354, 2019.

[161] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue : A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv :1804.07461*, 2018.

[162] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Superglue : A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[163] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.

[164] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, et al. Tacotron : Towards end-to-end speech synthesis. *Proc. Interspeech 2017*, pages 4006–4010, 2017.

[165] Z. Wang and A. Culotta. Identifying spurious correlations for robust text classification. *arXiv preprint arXiv :2010.02458*, 2020.

[166] B. Weber. Swift and slashing, computer topples kasparov. *The New York Times*. URL https://www.nytimes.com/1997/05/12/nyregion/swift-and-slashing-computer-topples-kasparov.html.

[167] P. J. Werbos. *The roots of backpropagation : from ordered derivatives to neural networks and political forecasting*, volume 1. John Wiley & Sons, 1994.

[168] J. Wiens and E. S. Shenoy. Machine learning for healthcare : on the verge of a major shift in healthcare epidemiology. *Clinical Infectious Diseases*, 66 (1) :149–153, 2018.

[169] I. H. Witten and D. N. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. *Wikipedia and Artificial Intelligence : An Evolving Synergy*, page 25, 2008.

[170] A. Wölker and T. E. Powell. Algorithms in the newsroom ? news readers' perceived credibility and selection of automated journalism. *Journalism*, 22 (1) :86–103, 2021.

[171] C. Wu, M. Tygert, and Y. LeCun. Hierarchical loss for classification. *arXiv preprint arXiv :1709.01062*, 2017.

[172] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv :1911.03814*, 2019.

[173] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*, 2016.

[174] B. Xu, Q. Wang, Y. Lyu, Y. Zhu, and Z. Mao. Entity structure within and throughout : Modeling mention dependencies for document-level relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14149–14157, 2021.

[175] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv :1601.01343*, 2016.

[176] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. Learning distributed representations of texts and entities from knowledge base. *arXiv preprint arXiv :1705.02494*, 2017.

[177] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto. Luke : deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv :2010.01057*, 2020.

[178] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection : A survey. *arXiv preprint arXiv :2110.11334*, 2021.

[179] Y. Yang, O. Irsoy, and K. S. Rahman. Collective entity disambiguation with structured gradient tree boosting. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, pages 777–786, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi : 10.18653/v1/N18-1071. URL https://www.aclweb.org/anthology/N18-1071.

[180] Y. Yao et al. Docred : A large-scale document-level relation extraction dataset. *arXiv preprint arXiv :1906.06127*, 2019.

[181] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv :2206.10789*, 2022.

[182] N. Zhang, X. Chen, X. Xie, S. Deng, C. Tan, M. Chen, F. Huang, L. Si, and H. Chen. Document-level relation extraction as semantic segmentation. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3999–4006. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi : 10.24963/ijcai.2021/551. URL https://doi.org/10.24963/ijcai.2021/551. Main Track.

[183] W. Zhou, K. Huang, T. Ma, and J. Huang. Document-level relation extraction with adaptive thresholding and localized context pooling. *arXiv preprint arXiv :2010.11304*, 2020.

université
PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies
de l'information et de
la communication (STIC)

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France