DeepType 2: Superhuman Entity Linking All You Need Is Type Interactions

Jonathan Raiman
Paris-Saclay, NVIDIA
Paris, France
jonathanraiman@gmail.com

Abstract

Across multiple domains from computer vision to speech recognition, machine learning models have been shown to match or outperform human experts at recognition tasks. We lack such a comparison point for Entity Linking. We construct a human benchmark on two standard datasets (TAC KBP 2010 and AIDA (YAGO)) to measure human accuracy. We find that current systems still fall short of human performance.

We present DeepType 2, a novel entity linking system that closes the gap. Our proposed approach overcomes shortcomings of previous type-based entity linking systems, and does not use pre-trained language models to reach this level. Three key innovations are responsible for DeepType 2's performance: 1) an abstracted representation of entities that favors shared learning and greater sample efficiency, 2) autoregressive entity features indicating type interactions (e.g. list type homogeneity, shared employers, geographical co-occurrence) with previous predictions that enable globally coherent document-wide predictions, 3) the entire model is trained end to end using a single entity-level maximum likelihood objective function. This is made possible by associating a context-specific score to each of the entity's abstract representation's sub-components (types), and summing these scores to form a candidate entity logit. In this paper, we explain how this factorization focuses the learning on the salient types of the candidate entities. Furthermore, we show how the scores can serve as a rationale for predictions.

The key contributions of this work are twofold: 1) we create the first human performance benchmark on standard benchmarks in entity linking (TAC KBP 2010 and AIDA (YAGO)) which will be made publicly available to support further analyses, 2) we obtain a new state of the art on these datasets and are the first to outperform humans on our benchmark. We perform model ablations to measure the contribution of the different facets of our system. We also include an analysis of human and algorithmic errors to provide insights into the causes, notably originating from journalistic style and historical context.

1 Introduction

Breakthroughs in natural language understanding from high-capacity language models with mask-based losses [Devlin et al. 2018] and pre-training on web-sized corpuses [Raffel et al. 2019] 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 (EL) similarly benefited from this wave of pre-trained language models [Logeswaran et al. 2019, Ling et al. 2020, Févry et al. 2020] where systems without task-specific features match the accuracy of those with EL features and structured data [Sil et al. 2018, Raiman and Raiman 2018]. Despite advances from novel architectures and pre-training, EL systems fall short of human performance with accuracies ranging from 90% to 96% on standard benchmark datasets [Raiman and Raiman 2018, Ling et al. 2020], while other NLP tasks such as sentiment-analysis [Raffel et al. 2019], named entity recognition [Yamada et al. 2020], or part of speech tagging rival human performance with accuracies above 97%.

Have we reached a performance ceiling on EL? We split this question into two parts: what is human performance on this task, and can we match it? We answer through two key contributions:

1. We establish a human performance benchmark on two frequently used standard Entity Linking datasets TAC KBP 2010 [Ji et al. 2010] (TAC) and AIDA (YAGO) [Hoffart et al. 2011] (AIDA) with annotations we make publicly available. 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.
2. We present DeepType 2, a new EL system that improves over the state of art (SoTA) on seven standard EL 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.

The paper is structured as follows: Section 2 states the EL problem; Section 3 presents related work; Section 4 describes our approach; Section 5 presents experiments measuring human EL accuracy and shows how DeepType 2 profits from type-based representations, negative sampling, and global normalization; Section 6 contains a discussion of the results, a conclusion, and future work directions.

2 Problem Statement

The goal of entity linking is to find the exact element (*entity*) in the knowledge base (KB) referenced by a pre-highlighted phrase (*mention*) in an input document[1]. Using an *alias table* collected on training corpora we can store potential mappings between mentions and entities. The task then becomes finding the correct entity among the alias table results, rather than considering all entities in the full KB (6+ million in the English Wikipedia). In Figure 1 we see that the alias table entry for “Ada" has two possible options: Ada Lovelace or Ada language.

3 Related Work

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

Abstract Entity Representations and Types. Ling, Singh, and Weld [2015] proposed to use the diverse types of NER tags (e.g., persons, places) to categorize all candidate entities in their EL system. The use of abstract entities was further generalized in DeepType [Raiman and Raiman 2018], 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 [Nie et al. 2018, Logeswaran et al. 2019]. In [Mulang’ et al. 2020], the proposed EL 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 [Gutmann and Hyvärinen 2010]. 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 [Nie et al. 2018].

Coherency. EL selects entities based on their individual relevance, where a key component is their compatibility with the other document entities. The connections between entities can be measured using reciprocal link statistics from Wikipedia [Milne and Witten 2008], by analyzing the link graph using a PageRank algorithm [Pershina, He, and Grishman 2015], or by learning a distributed representation of entities that captures co-occurence [Yamada et al. 2016]. As the number of potential entity pairs is large, computing coherence metrics presents a computational challenge. In Globerson et al. [2016] the authors use attention over a subset of the document mentions to reduce the computational cost.

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 [Nie et al. 2018, Wu et al. 2019, Mulang’ et al. 2020, Ling et al. 2020]. 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 (EL). The high memory and computation cost limit the applicability of these models to long documents. The current SoTA [Févry et al. 2020] 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.

Discussion. Several lessons are learned from the above approaches. For instance, rich classes as in DeepType [Raiman and Raiman 2018] enable a type Oracle to reach 99% accuracy on TAC and AIDA, but their type classifier does not realize this potential. Furthermore, this approach forces to coerce entities into exclusive type labels, which requires human intervention in this preprocessing step. Along these lines, our proposed approach associates entities with a variable number of type labels removing preprocessing. We switch the objective function from predicting types to entity disambiguation in a contrastive loss setting. To reduce the computational cost of coherency metrics, pairwise entity features are materialized by DeepType 2 during search by live querying a KB.

4 Approach

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 2.

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 [Graves and Schmidhuber 2005] (1 in Figure 2).

Mention Representation For each mention we use an *alias table* to generate candidate entities. Our alias table is generated using the same approach as prior work [Ferragina and Scaiella 2010]: intra-wiki links from Wikipedia provide a mapping from mention to linked entities. We also exploit the link statistic features from the alias table: 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.

[1] Our work focuses on EL, which differs from end-to-end EL where the task also involves mention detection.

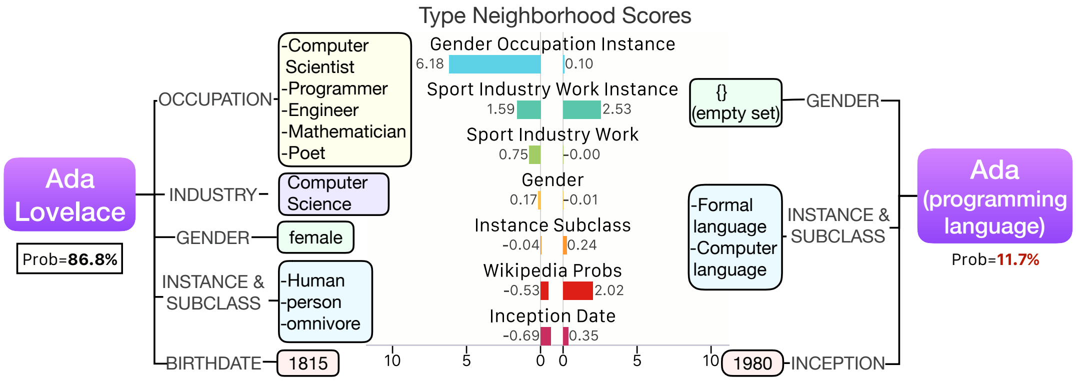

Figure 1: Disambiguating “Ada" in the sentence “Ada wrote the first computer program. She..." Type neighborhoods for candidate entities are computed by finding depth 2 neighbors via different typed Wikidata edges. An entity’s score is the sum of its type neighborhood and interaction scores. This acts as a rationale for DeepType 2’s decisions. We see wikipedia probs, gender, occupation, instance, and work had the largest impact.

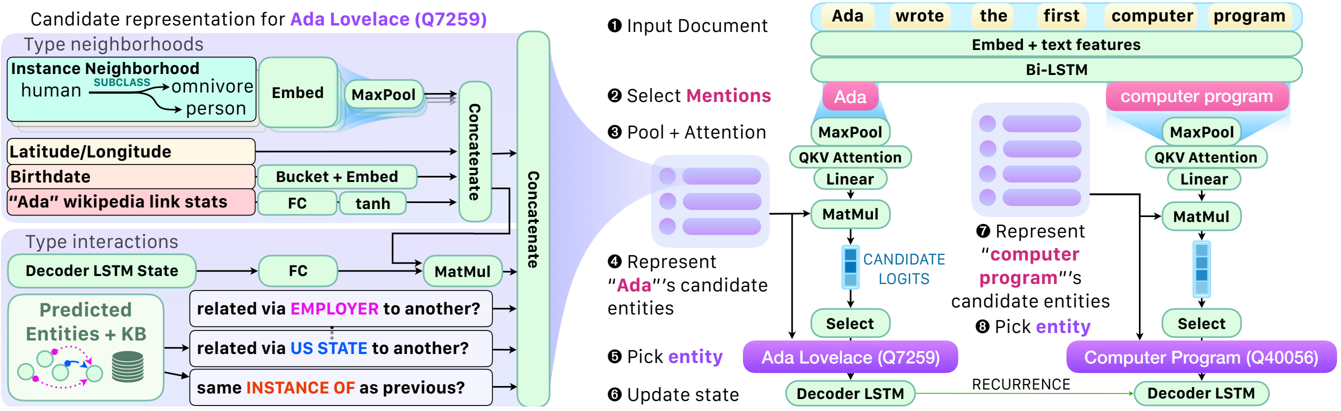

Figure 2: An LSTM reads text, while a separate graph NN produces candidate entity representations used for prediction. Entity predictions are fed to a Decoder LSTM. The decoder LSTM and predicted entities produce type interaction features for future predictions.

We obtain for each document-mention pair D_m a fixed-length representation $h_m(D_m)$ from the variable number of mention tokens. First we max-pool the associated Bi-LSTM hidden states (3 in Figure 2) to produce $h_{\text{pool},m}(D_m)$. Second, we obtain longer-range context using QKV Attention [Vaswani et al. 2017] ($\text{Att}(\cdot)$) over the full document with $h_{\text{pool},m}(D_m)$ as query, and linearly project the pooled and attended vector into the same space via a learnt matrix W to obtain h_{m}:

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

Entity Representation 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 Appendix D 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 $n_{neighborhood\,depth}$ steps away as the *type neighborhood* representation of an entity. See Figure 1 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 $n_{neighborhood\,depth} = 2$, which enables us to 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 [Wang et al. 2019].

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 (6. in Figure 2). 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 5, certain relations are redundant, and the system is robust to removing those. See Appendix D 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 3 an example of these interactions to disambiguate John Gorst.

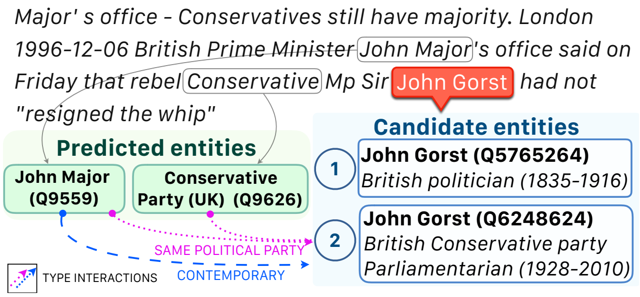
Major' s office - Conservatives still have majority. London
1996-12-06 British Prime Minister John Major's office said on
Friday that rebel Conservative Mp Sir John Gorst had not
"resigned the whip"
Predicted entities
John Major (Q9559)
Conservative Party (UK) (Q9626)
Candidate entities
1
John Gorst (Q5765264)
British politician (1835-1916)
2
John Gorst (Q6248624)
British Conservative party Parliamentarian (1928-2010)
SAME POLITICAL PARTY
CONTEMPORARY
TYPE INTERACTIONS

Figure 3: 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.

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 $f_t(c_i, D_m, s)$ the concatenation of type neighborhood and interaction features:

$$\text{Score}(c_i, D_m, s) = h_m(D_m) \cdot f_t(c_i, D_m, s), \tag{2}$$

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

Rewriting $\text{Score}(c_i, D_m, s)$ as a sum of feature scores (see Appendix E) reveals each type neighborhood or interaction's contribution to the overall score. Feature scores may serve as decision justifications as we show in Figure 1.

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:

$$\mathcal{L}(\theta) = \sum_{\{e, D_m, s\}} -\log \mathbb{P}(e | D_m . s; \theta). \tag{4}$$

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 [Gutmann and Hyvärinen 2010]. 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. $\sim 2^{128}$ in DeepType [Raiman and Raiman 2018]) most of which are impossible. A contrastive loss focuses the learning on discriminative features: the gradient is zero for features common between candidates (proof in Appendix F).

Densification

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 2.96x from 74M to 220M mentions. Details for our classifier are given in Appendix A.

Coherency

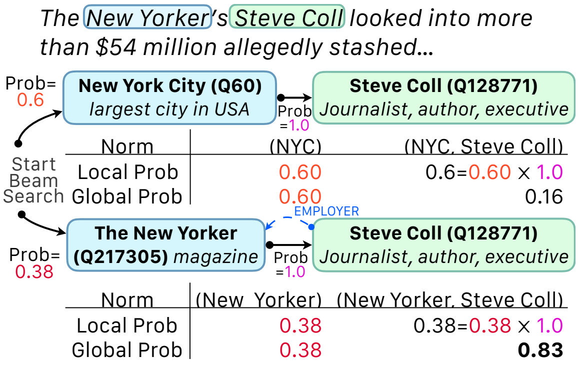

Figure 4: Global normalization effect in TAC: Steve Coll, although unambiguous, reinforces the likelihood of picking his employer New Yorker magazine when scores are summed before being normalized.

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.
- Beam search with autoregressive features 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 4. This conversion from local to global was previously proposed to improve expressivity and overcome label bias [Andor et al. 2016, Raiman and Miller 2017], an autoregressive model pitfall [Lafferty, McCallum, and Pereira 2001].

5 Results

We first seek to identify the remaining gap between humans and algorithms on EL by establishing a human performance benchmark. Second, we evaluate DeepType 2 on standard benchmarks. Third, 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 [Pershina, He, and Grishman 2015], otherwise our alias table is built from intra-wiki links. Neural network hyperparameters were selected using a Wikipedia-based validation set and are given in Appendix C. Our code is available at this url[2]. The Wikipedia dump is downloadable at this url[3] and is licensed under CC BY-SA and GFDL. AIDA's train split is freely available for research purposes from NIST[4].

Human performance benchmark

We measure human performance using a panel of annotators from Amazon Mechanical Turk (AMT) on TAC and AIDA. We find that humans reach 96.86% on TAC and 96.78% on AIDA, outperforming the current state of the art on these tasks as shown in Table 2.

To improve response quality we take particular care to screen, brief, and provide proper incentives to the annotators: 1) they must have AMT's Master qualification, a recognition of prior excellence in annotation tasks, 2) we give a bonus for correct answers to align incentives, 3) annotators are screened by testing that they read the instructions and requiring they reach a minimum accuracy on a trial subset of the data. To further reduce the effect of annotator expertise differences, each labelled mention is assigned to 3 different annotators and we measure accuracy using the best provided response (Oracle). Inter-annotator agreement (Fleiss's κ) is high, supporting the belief that annotators reached similar conclusions and did not respond randomly. A summary of our results are given in Table 1, details of the AMT annotation interface are given in Appendix B, and we release the annotations from this benchmark at this url[5].

Table 1: Overall statistics from human performance benchmark.

	TAC	AIDA
Oracle acc. (%)	96.86	96.78
Majority acc. (%)	95.39	93.35
Agreement (Fleiss' κ)	96.84	93.96
Mean response time (seconds)	18.49	15.68

Table 2: Humans and state of the art EL system accuracy ($\mu \pm \sigma$).

Model	TAC	AIDA
Human Oracle	96.86	96.78
DeepType 2 (ours)	**97.48**±0.06	**97.72**±0.04
Ling et al. [2020]	89.8	94.9
Raiman and Raiman [2018]	90.85	94.88
Mulang' et al. [2020]	-	94.94
Févry et al. [2020]	94.9	96.7

Evaluation on Standard Datasets

Table 3: EL system accuracy on standard datasets ($\mu \pm \sigma$).

Dataset	DeepType 2 (ours)	Yang et al. (2018)	De Cao et al. [2020]
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

We compare Human performance, DeepType 2, and the current EL 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 2. In Table 3 we report evaluations of our system on five additional well known EL datasets WNED-WIKI [Guo and Barbosa 2018], WNED-CWEB [Guo and Barbosa 2018], MSNBC [Cucerzan 2007], AQUAINT [Milne and Witten 2008], and ACE 2004 [Ratinov et al. 2011].

We first find that humans outperform existing approaches, suggesting that there is room for algorithms to improve. Humans have similar accuracy on TAC and AIDA, while surprisingly SoTA algorithmic approaches until 2020 perform 4.09% higher on AIDA than TAC. DeepType 2 improves

[2] Anonymized during review

[3] https://dumps.wikimedia.org

[4] https://trec.nist.gov/data/reuters/reuters.html

[5] Anonymized during review.

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%).

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%.

Table 4: Impact ($\mu \pm \sigma$) of decision method on accuracy.

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 5: Impact ($\mu \pm \sigma$) of varying search beams k on accuracy.

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

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 4. 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 5. We find that a small percentage of search errors in TAC and AIDA can be mitigated by considering more hypotheses.

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 6.

Table 6: Typed confusions for DeepType 2 (DT2) and humans (H).

Confusion	TAC (%)		AIDA (%)	
	DT2	H	DT2	H
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

Ablations

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 7. We empirically verify the effect of replacing type neighborhoods by same dimension unique Entity-Vectors (EV) used in SoTA approaches [Yamada et al. 2016, Sil et al. 2018, Ling et al. 2020, Févry et al. 2020]. Type neighborhoods have 6 times less parameters (166M vs. 998M), get the same accuracy as EV 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 EV system's performance.

Table 7: Impact of entity representation on accuracy.

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.

Type interaction features 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 5. 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 5. "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.

Training ablations

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 8.

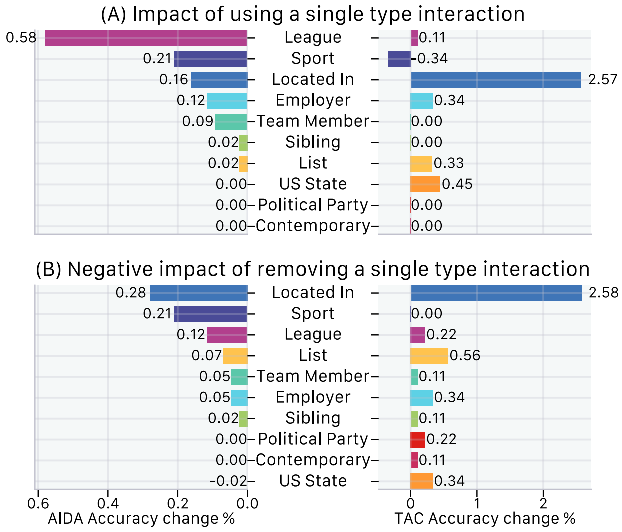

Figure 5: Type interactions are domain dependent as visible in (A) by looking at the impact of using a single relation in TAC vs. AIDA. In (B) we test the redundancy of type interactions by removing one from the system.

Table 8: Impact of Wikipedia Densification on accuracy.

Training data	TAC	AIDA
Original	96.75	96.47
Densified	97.48	97.72

Negative sampling As entity representations are only learnt through comparisons, the unambiguous mentions provide no supervision potentially leaving representations untrained. In Table 9 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.

6 Discussion and Conclusion

We establish an Entity Linking human benchmark that measures human performance and provides a milestone for algorithmic approaches. The principal difficulty in setting up this benchmark is to obtain high quality responses from annotators. We use trial runs and bonuses for correct answers to obtain high inter-annotator agreement and accurate responses. We further reduce the effect of annotator expertise differences by measuring the best human response (oracle). Through this benchmark, we observe that previous systems approach human performance but still underperform.

We close the performance gap thanks to a new EL 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 EL 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 [Raiman and Raiman 2018]: 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.

Table 9: Negative sampling impact on EL performance.

Negative Samples	Max candidate entities/mention if training	TAC	AIDA
0	20	95.41	95.63
20	20	96.98	97.99
100	100	97.48	97.72

Our work has several limitations. First, there is a need to extend the human benchmark by broadening it to additional datasets and languages. Second, DeepType 2 relies solely on structured relations and cannot make use of the wealth of unstructured relations. Third, 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.

References

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

Cucerzan, S. 2007. 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)*, 708–716.

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

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

Ferragina, P.; and Scaiella, U. 2010. 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*, 1625–1628. ACM.

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

Globerson, A.; Lazic, N.; Chakrabarti, S.; Subramanya, A.; Ringaard, M.; and Pereira, F. 2016. 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, 621–631.

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

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

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

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

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

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

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

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

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

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

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

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

Pershina, M.; He, Y.; and Grishman, R. 2015. 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*, 238–243.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text Transformer. *arXiv preprint arXiv:1910.10683*.

Raiman, J.; and Miller, J. 2017. Globally Normalized Reader. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1059–1069.

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

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

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

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

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

Wu, L.; Petroni, F.; Josifoski, M.; Riedel, S.; and Zettlemoyer, L. 2019. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. *arXiv preprint arXiv:1911.03814*.

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

Yamada, I.; Shindo, H.; Takeda, H.; and Takefuji, Y. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. *arXiv preprint arXiv:1601.01343*.

Yang, Y.; Irsoy, O.; and Rahman, K. S. 2018. 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)*, 777–786. New Orleans, Louisiana: Association for Computational Linguistics.