A Densifying Wikipedia Links

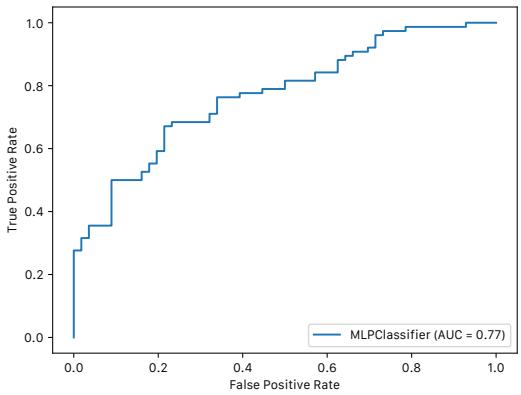

Figure 6: ROC Curve for the synthetic link classifier.

Table 10: Wikipedia corpus augmentation statistics

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

Approach

We increase the number of links in wikipedia to compensate for the lack of repeat and self-links. A similar technique was employed in DAWT to [Spasojevic et al. 2017] to increase the number of links by 4.8x to obtain a mention detection and entity co-occurence dataset, but the effect on entity linking was not yet tested. To increase the frequency of mentions in Wikipedia pages and increase exposure to repeat and self-links we train a binary classifier as our filter using 400 manually collected labels with a 2:1 train test split. For each potential additional link to an external page the classifier receives as features: number of links from the current page to the external page, overall link statistics for the external page, whether the proposed mention in the current page already links to the external page. The classifier is a 2-layer MLP with 5 hidden dimensions, a ReLu nonlinearity, and trained with the Adam optimizer [Kingma and Ba 2014]. The classifier has an accuracy of 71%, and its ROC curve is shown in Figure 6. The effect of densification on the training corpus is reported in Table 10. Along with this paper we will publish our classifier weights and training data.

Table 11: Impact of Wikipedia Densification on NLL.

	Wikipedia NLL	
Training data	Original	Densified
Original	0.73	1.13
Densified	0.81	0.84

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 11. We see a small change in NLL 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 NLL 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.

B Human Evaluation

Annotation interface

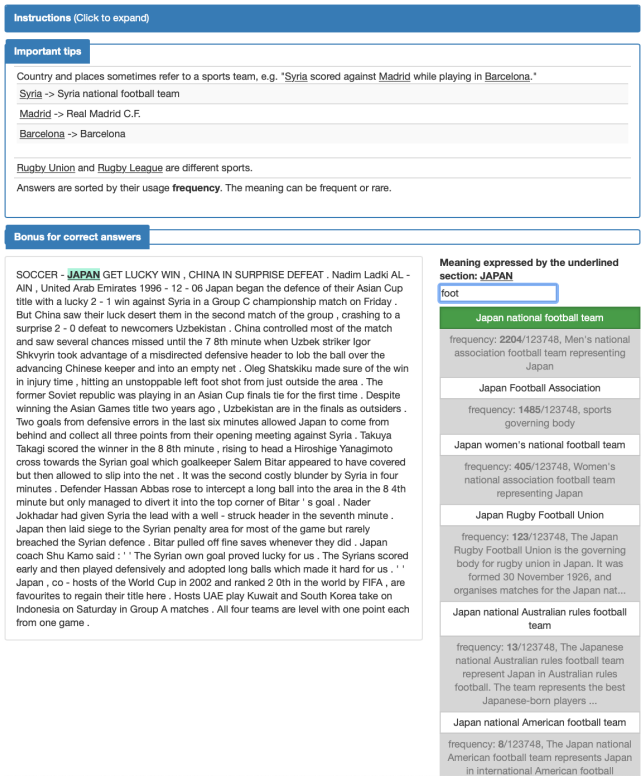

Figure 7: The annotation interface in Amazon Mechanical Turk shows a single highlighted mention at a time. Options are shown in a list with descriptions, title, and link frequency stats. To assist annotators the results are ordered by link frequency and a full-text search bar enables quick filtering 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.

Each labelled mention is presented to humans by highlighting the mention within the full original document. Certain documents contain multiple mentions (in AIDA there are on average 15 per document) but we show mentions in isolation to reduce confusion. Fortunately, document reading time is amortized because all the mentions from the same document

are offered to the annotator consecutively. Annotators select candidate entities from a list generated given by a Wikipedia alias table[6]. 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 7.

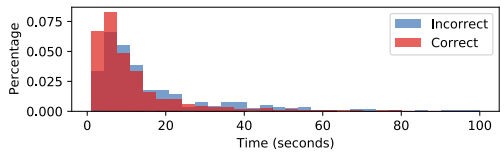

Figure 8: Time taken by AMT annotators grouped by correct and incorrect response times.

Response time 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 8. 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.

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.

C Neural Network Hyperparameters

DeepType 2's neural network dimensions and learning rate schedule were selected using a Wikipedia-based validation set and are provided in Table 12. We use the Adam optimizer [Kingma and Ba 2014]. 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.

D Entity Representation

Type Neighborhoods

Type neighborhoods are constructed using the relations given in Table 13 and use the embedding dimensions in Table 14. We also construct type neighborhoods that are a combination of these neighborhoods in Table 15. 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 15.

[6]The alias table always contains the correct answer.

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 12: Neural Network Hyperameters

Table 13: Wikidata relations for each type neighborhood.

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

Type interactions

We define type interactions using Wikidata relations between entities. We present the exact relations used in Table 16.

We also prune candidates when a pair of mention are connected by a specific syntactic pattern and a pair of their candidates could obey typed interaction. These patterns are given in Table 17 with syntax in bold.

E Feature scores

The feature-vector $f_t(c_i, D_m, s)$ is formed by concatenating $I_0(c_i), \ldots, I_F(c_i)$. By extracting the part of h_m that will be multiplied with feature $I_j(c_i)$ using a function $\text{Slice}(h_m, I_j(c_i))$, we recover feature scores:

Table 14: Type neighborhood used to represent entities.

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 15: Type neighborhoods with cross-terms.

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

$$\mathrm{Score}(c_i, D_m, s) = \sum_{j=0}^{F} \underbrace{\mathrm{Slice}(h_{\mathrm{m}}(D_m), I_j(c_i, s)) \cdot I_j}_{\text{feature } I_j\text{'s score}}, \tag{5}$$

$$= \sum_{j=0}^{F} \mathrm{Score}_{I_j}(c_i, D_m, s). \tag{6}$$

F Contrastive Loss

The likelihood of a candidate entity is computed by computing an exponential normalization ($\mathrm{Softmax}$) 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. We provide a proof below:

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

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

(8)

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

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

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

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

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

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.

Table 16: Wikidata relations for each type interaction.

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 17: Wikidata relations used within syntactic patterns.

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

G Ethical Considerations

Measuring human performance presents ethical and editorial challenges. First, the annotation task requires humans to select a single entity for each mention, but the answer might be unavailable, controversial, or too subjective. Therefore the results from our benchmark should not be seen as a perfect measure of cognitive skill, but rather as also a measure of distance or agreement with the pre-existing TAC and AIDA labels.

Because our presented system DeepType 2 uses the Wikidata knowledge graph and is trained on Wikipedia and news corpora, its representation of the world will carry along potential biases from these datasets. Wikipedia in particular has higher coverage of English and Anglosaxon content, and links more frequently point to historical content. The system should therefore be retuned when adapting it to other non encylopedic or news settings. As the system enables automated analyzis of natural language, it can be used by state-level actors to detect or track specific keywords and phrases.

Appendix References

Nemanja Spasojevic, Preeti Bhargava, and Guoning 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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.